

Figure 1

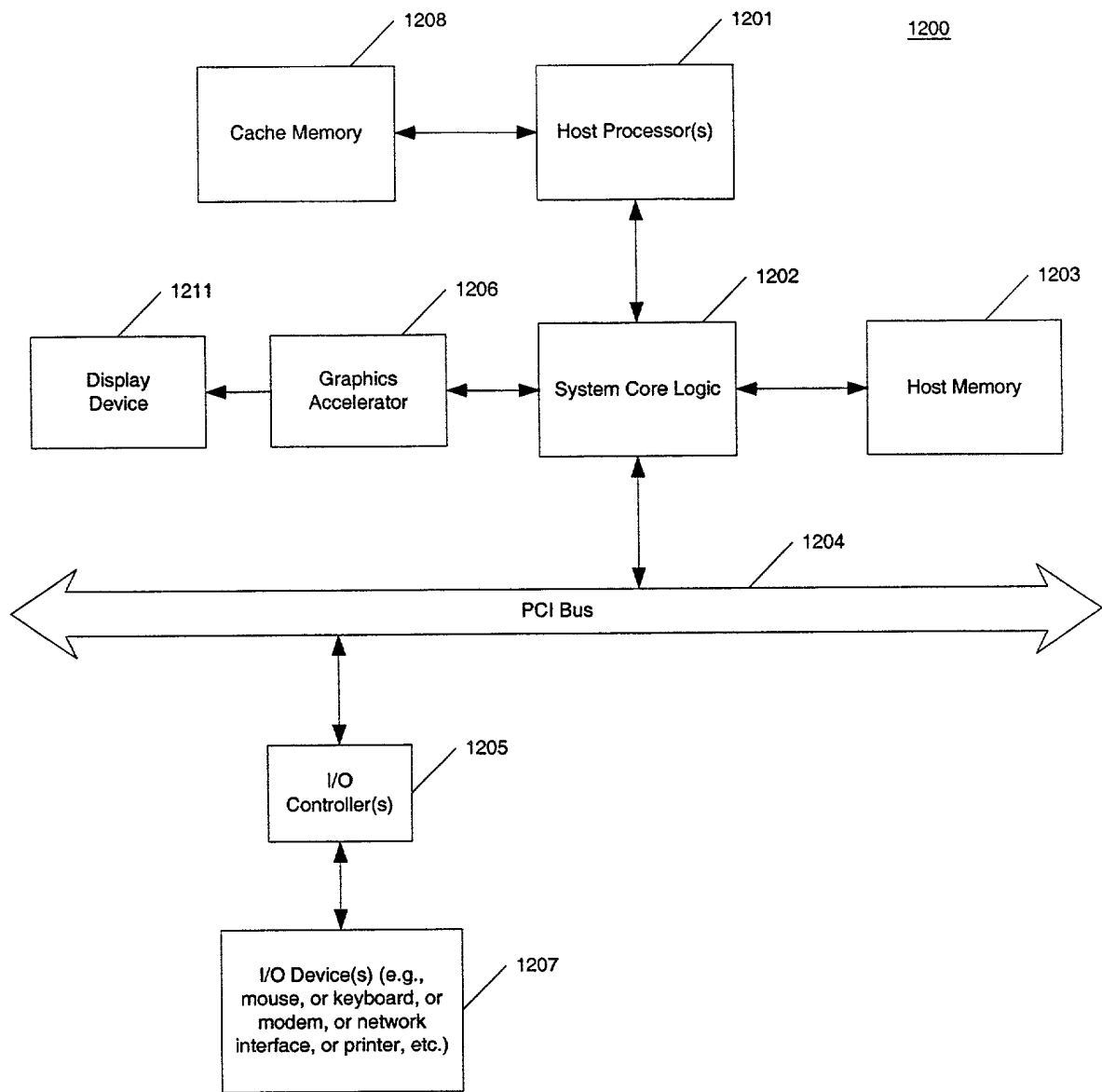


Figure 2

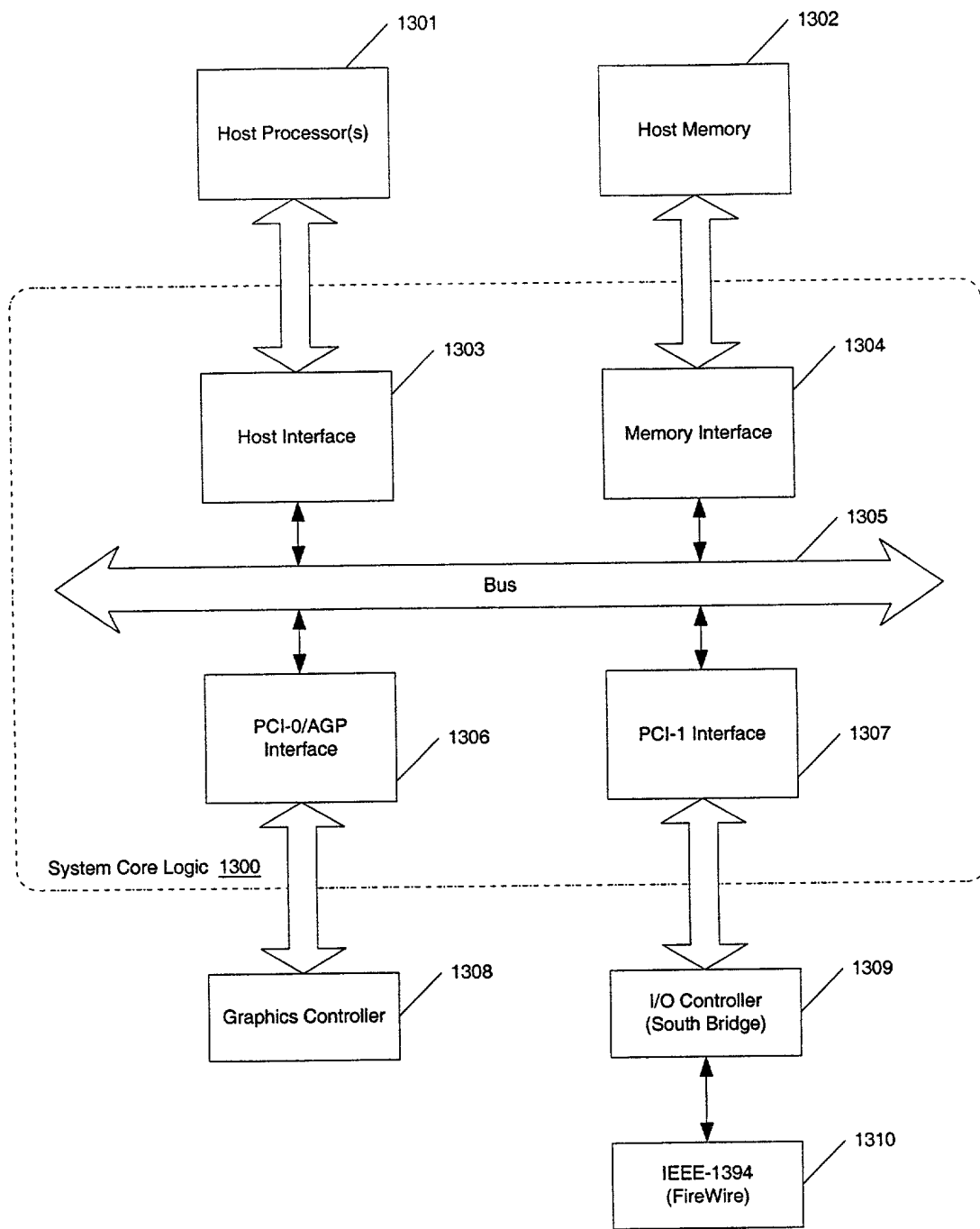


Figure 3

1400

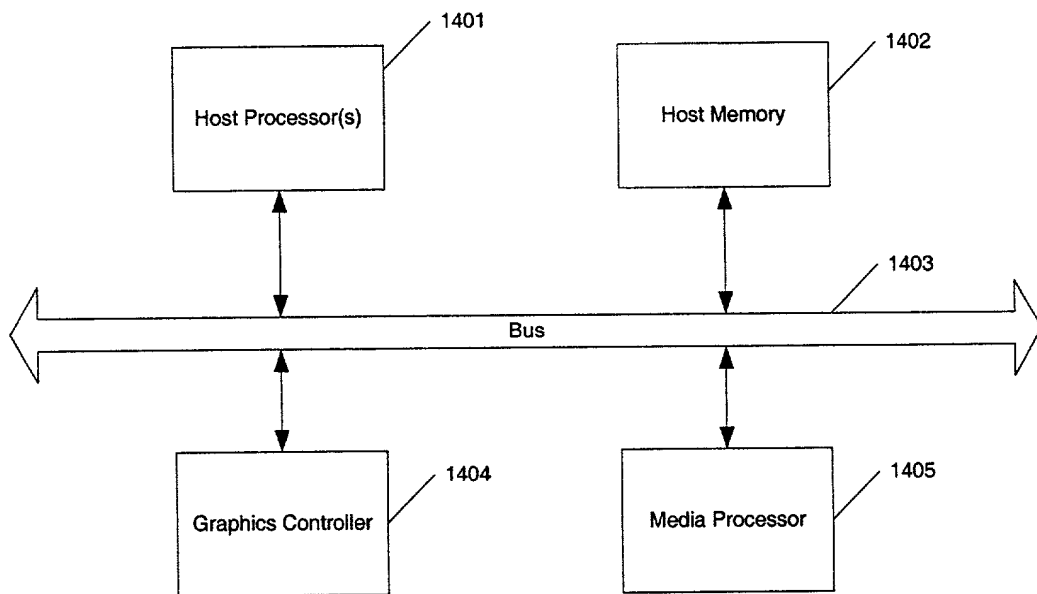


Figure 4A

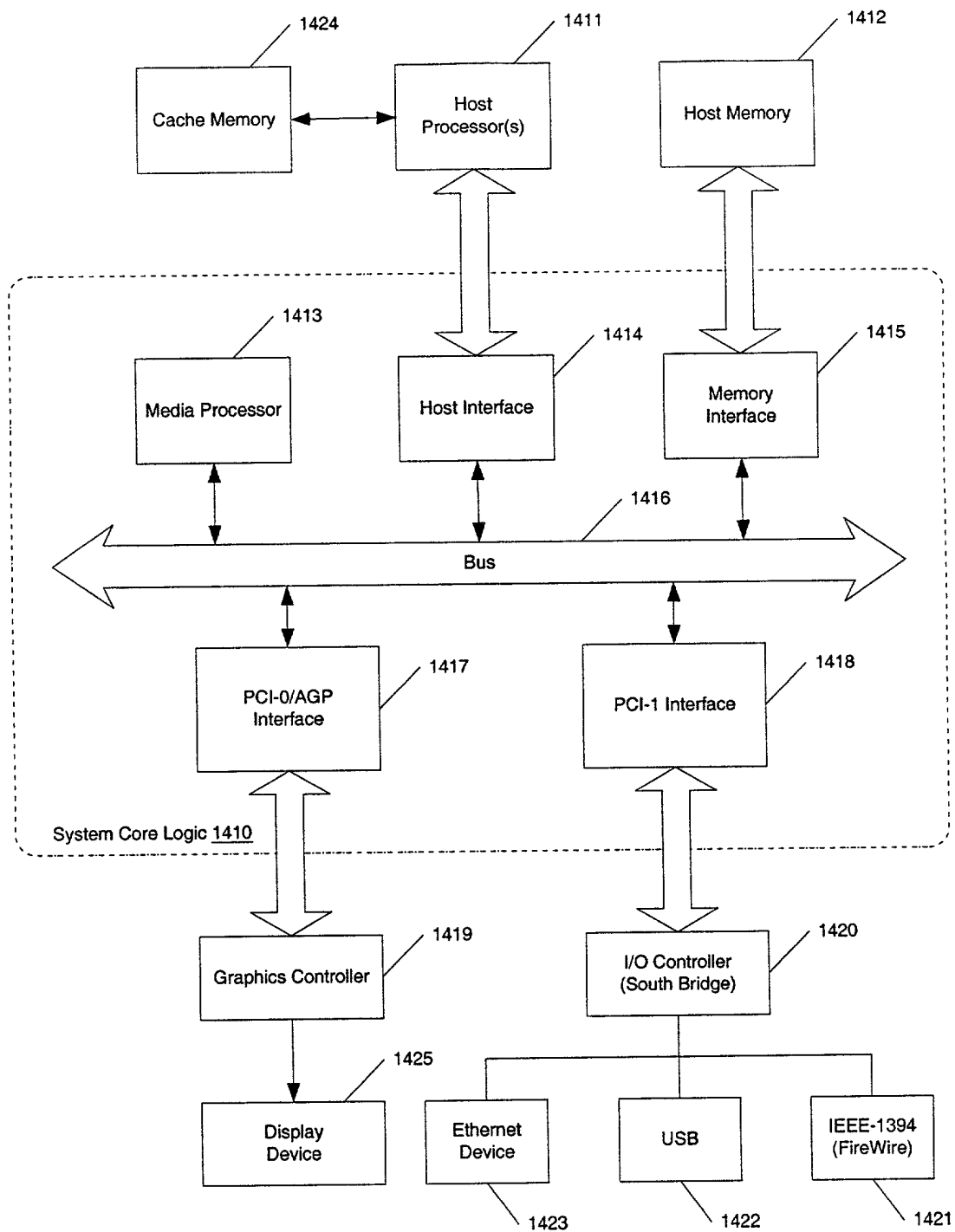


Figure 4B

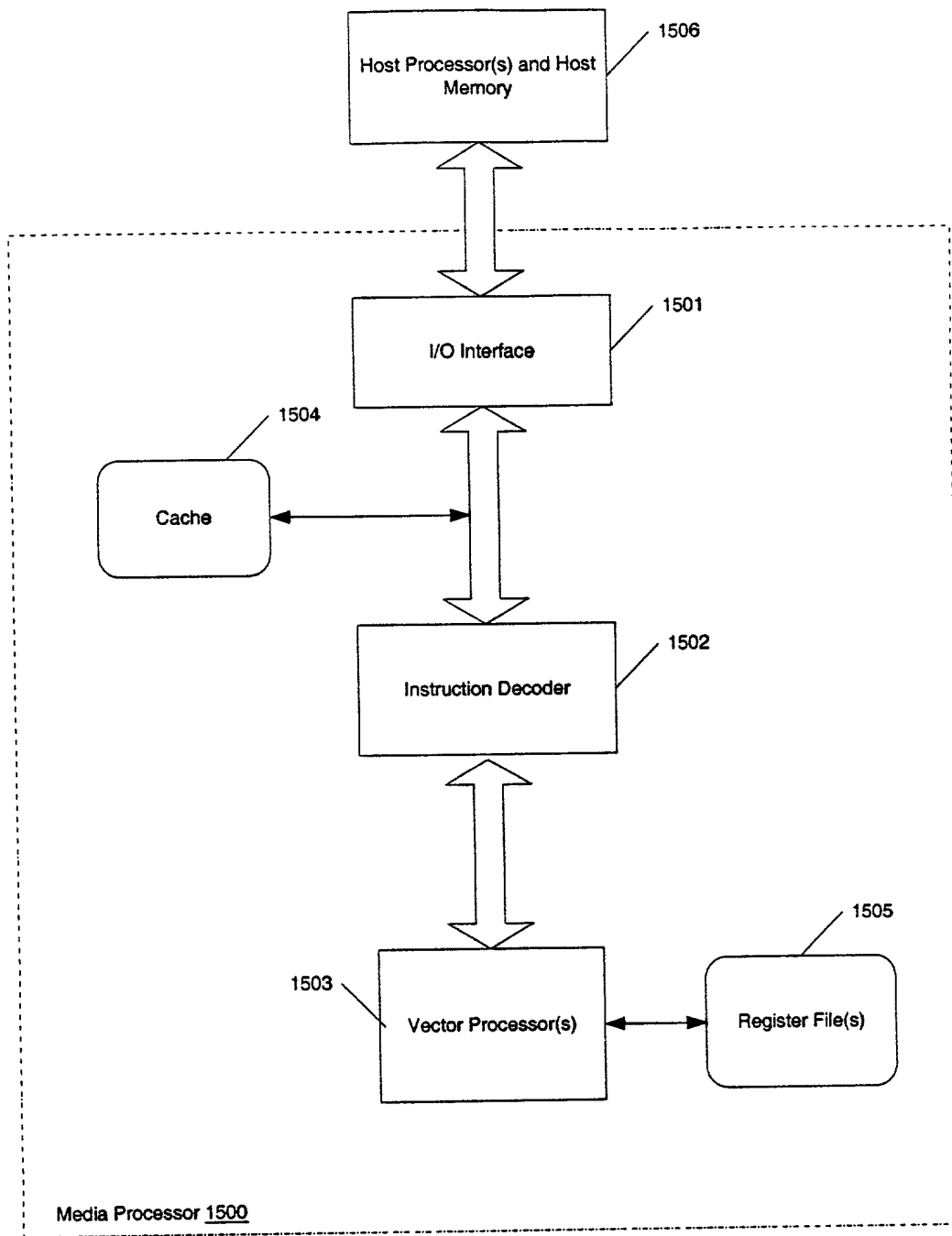


Figure 5A

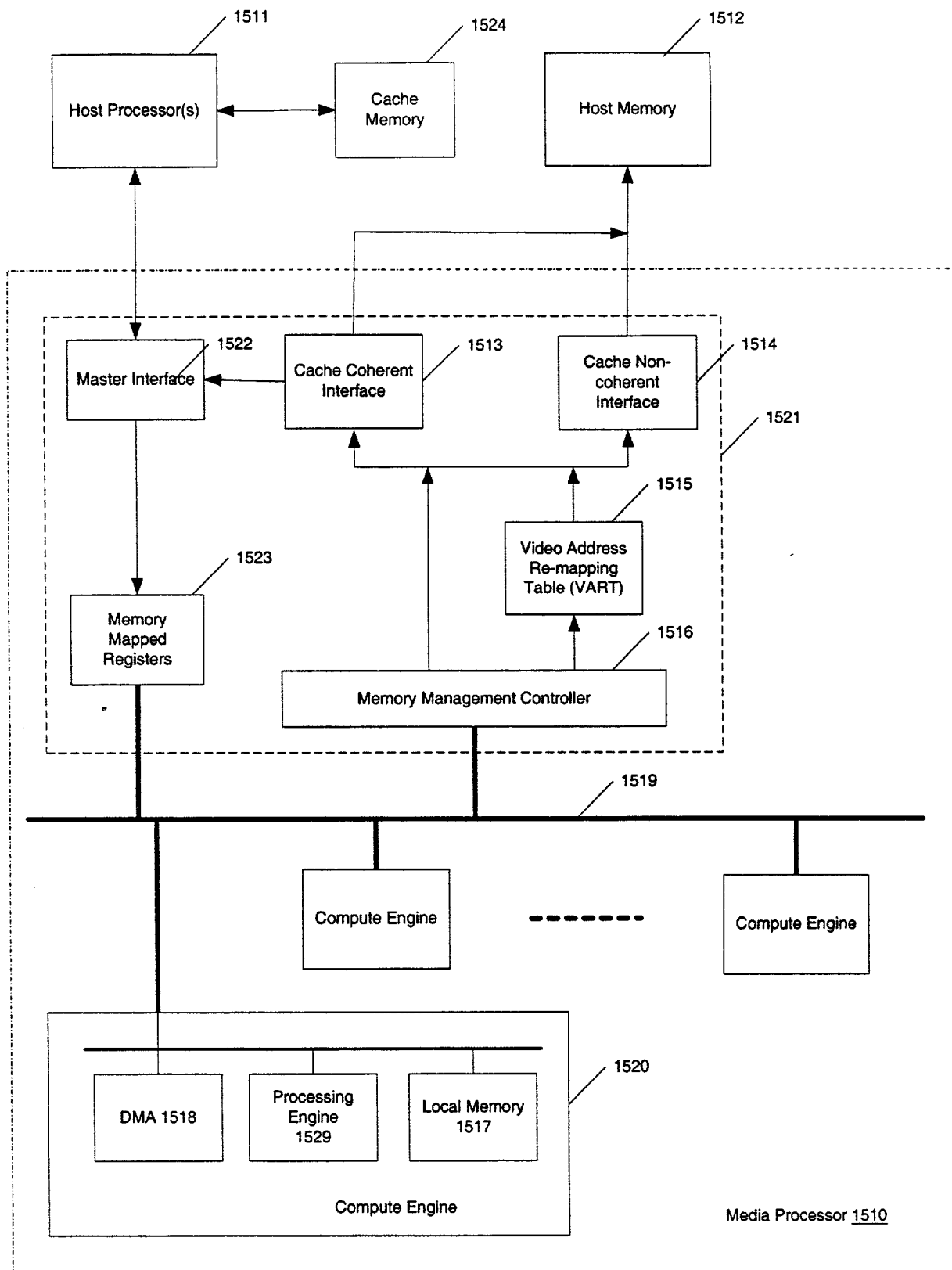


Figure 5B

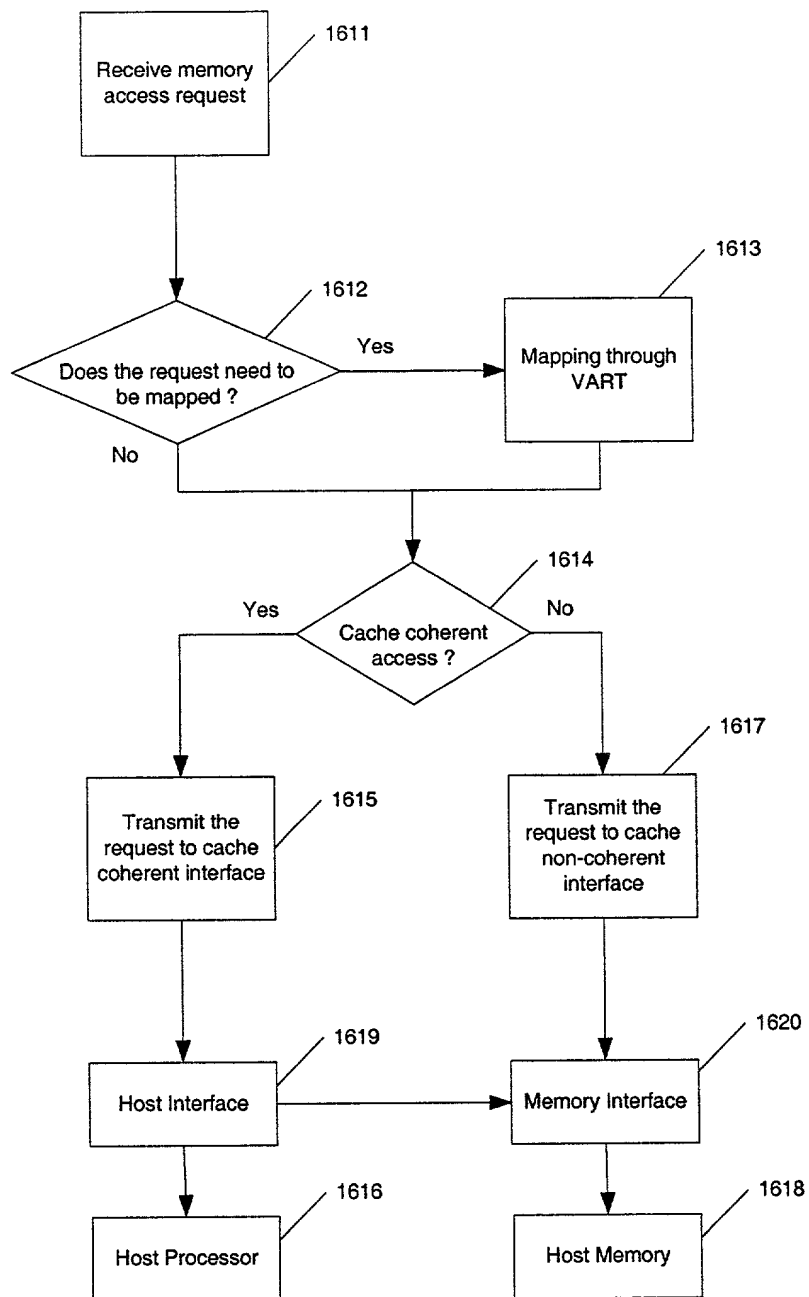
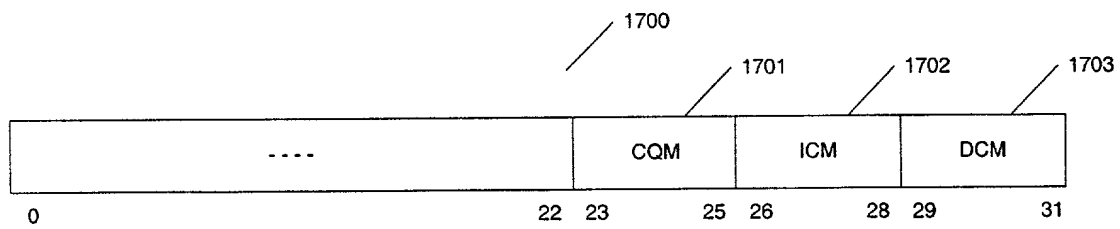


Figure 6



| Memory Access Mode Code | |
|-------------------------|--|
| Code | Description |
| 0 - - | Mapped |
| 100 | Unmapped and coherent |
| 101 | Unmapped and non-coherent |
| 110 | if (LogicalAddress[0] = 0) then mapped else unmapped and coherent |
| 111 | if (LogicalAddress[0] = 0) then mapped else unmapped and non-coherent |

Figure 7

1800

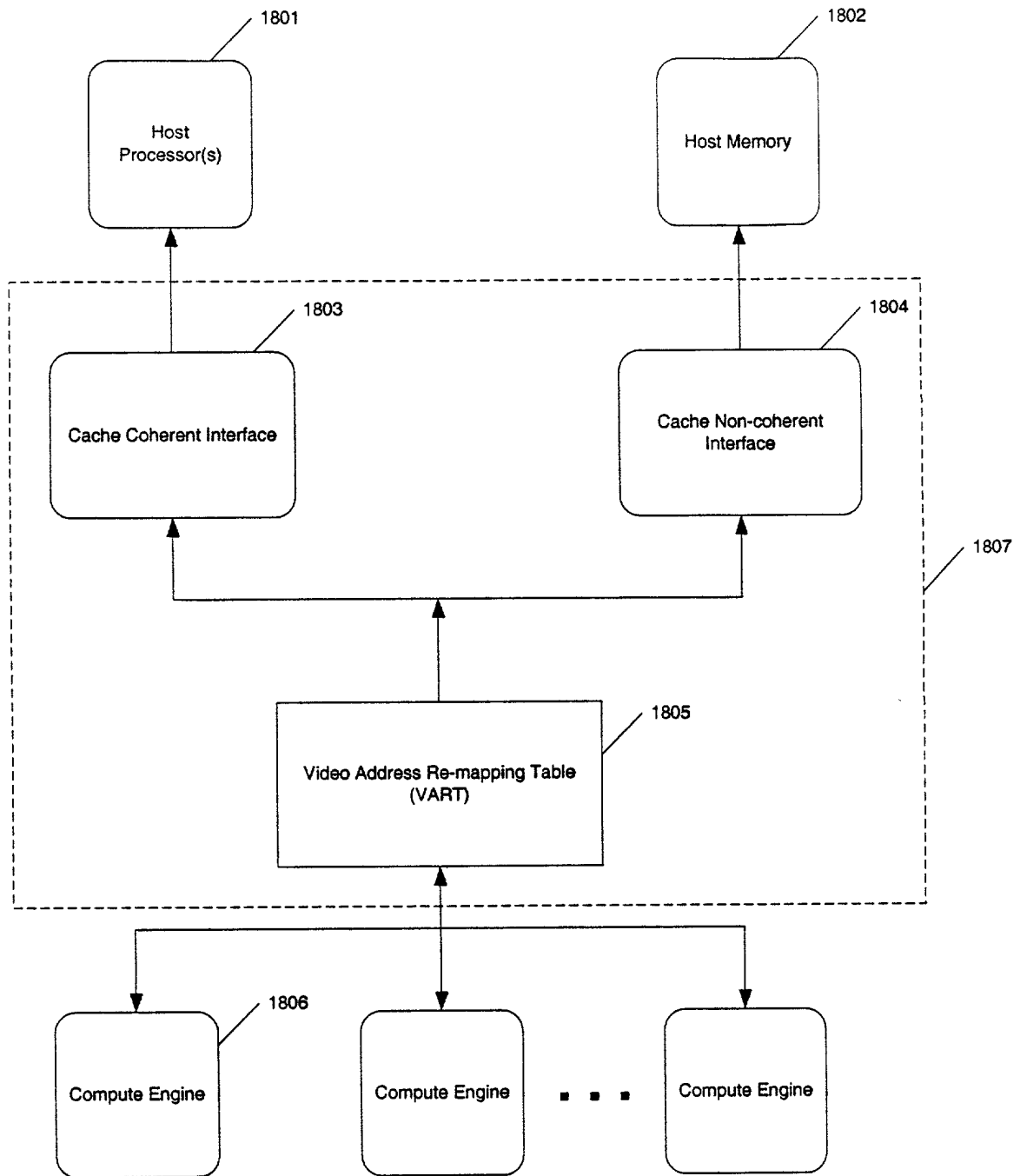


Figure 8

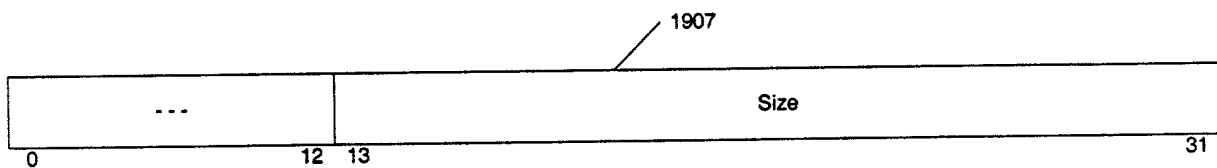
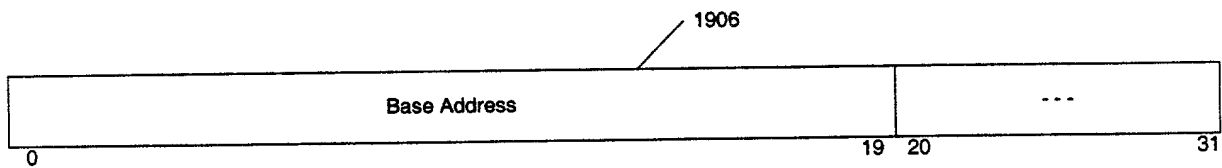
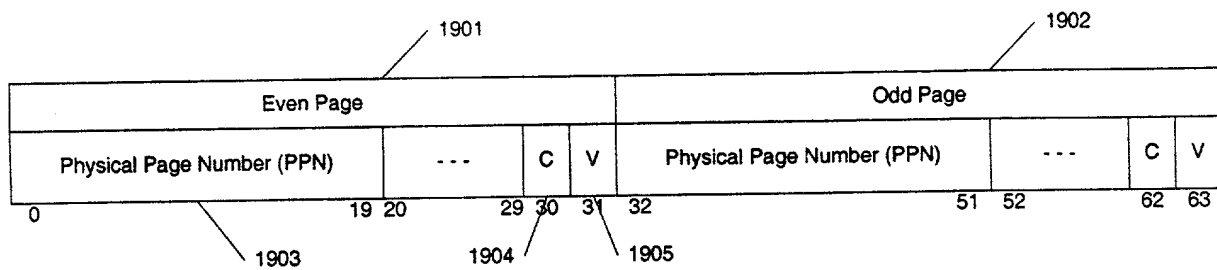


Figure 9

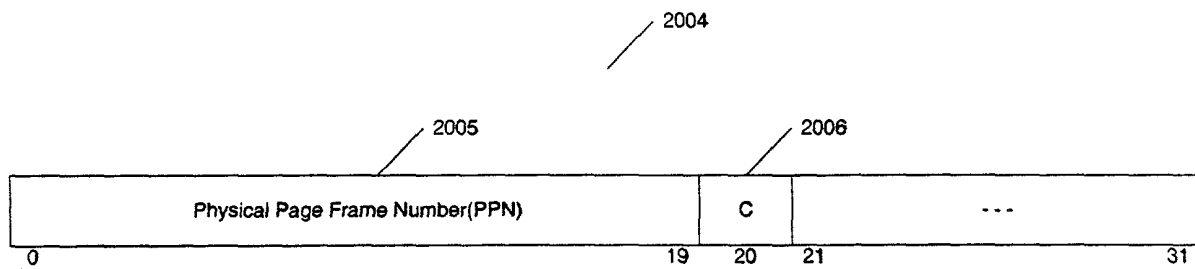
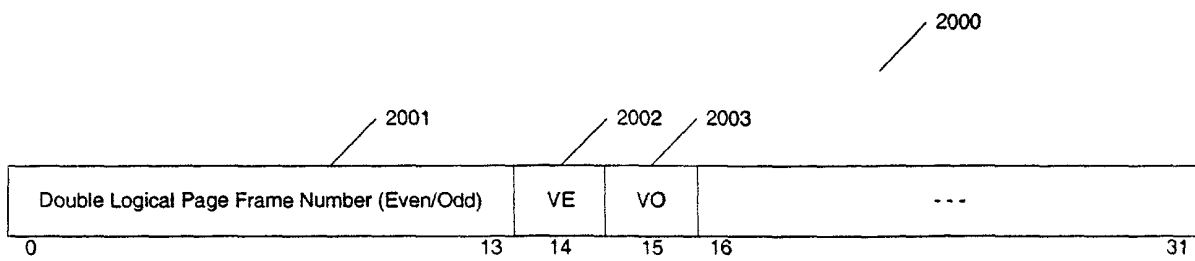


Figure 10A

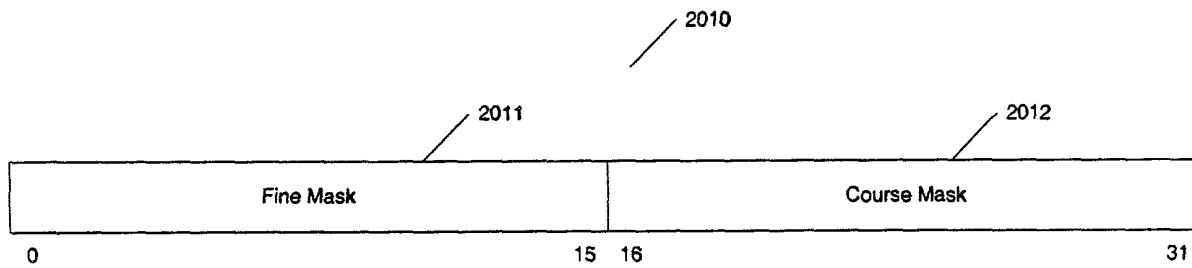


Figure 10B

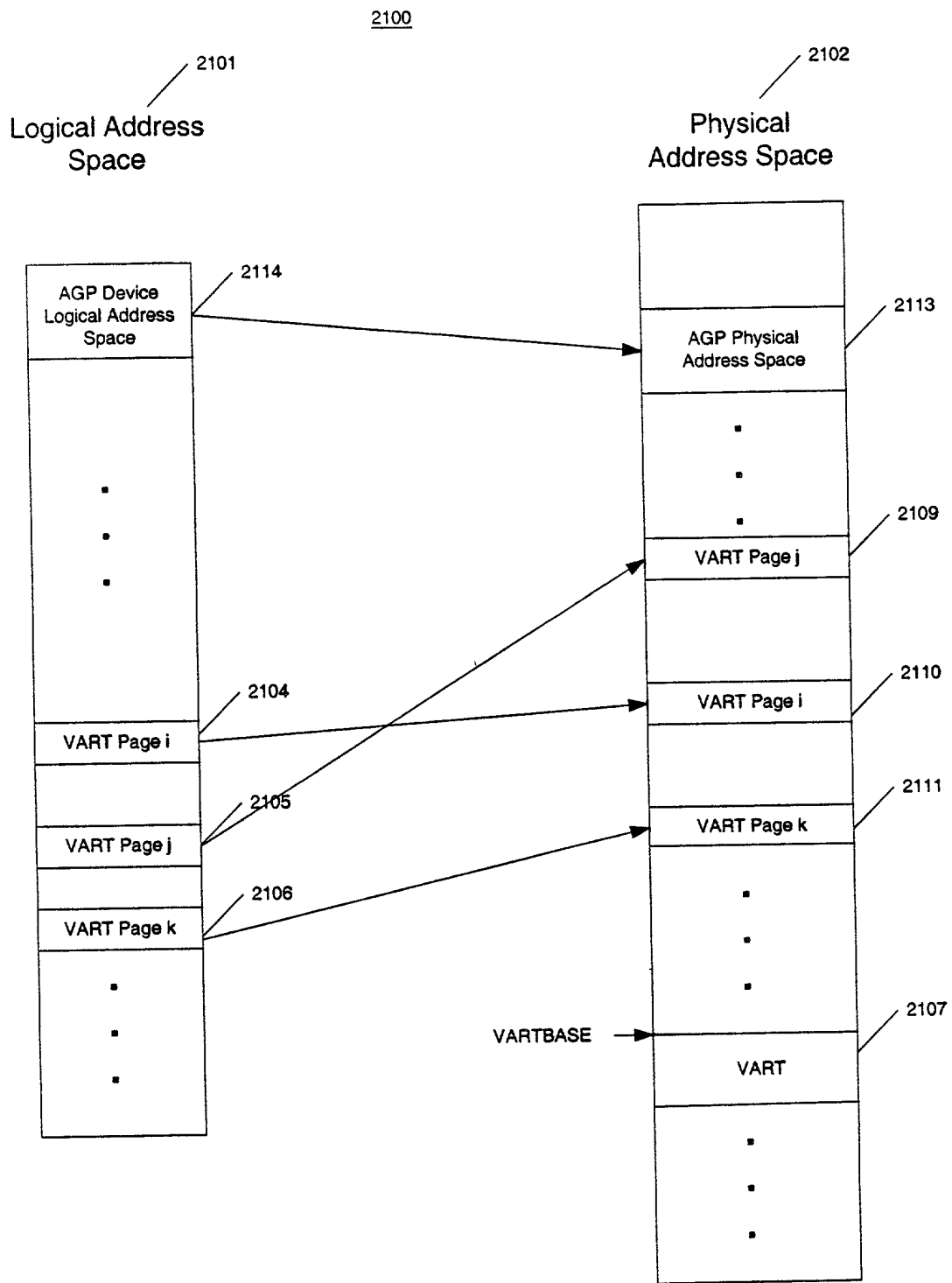


Figure 11

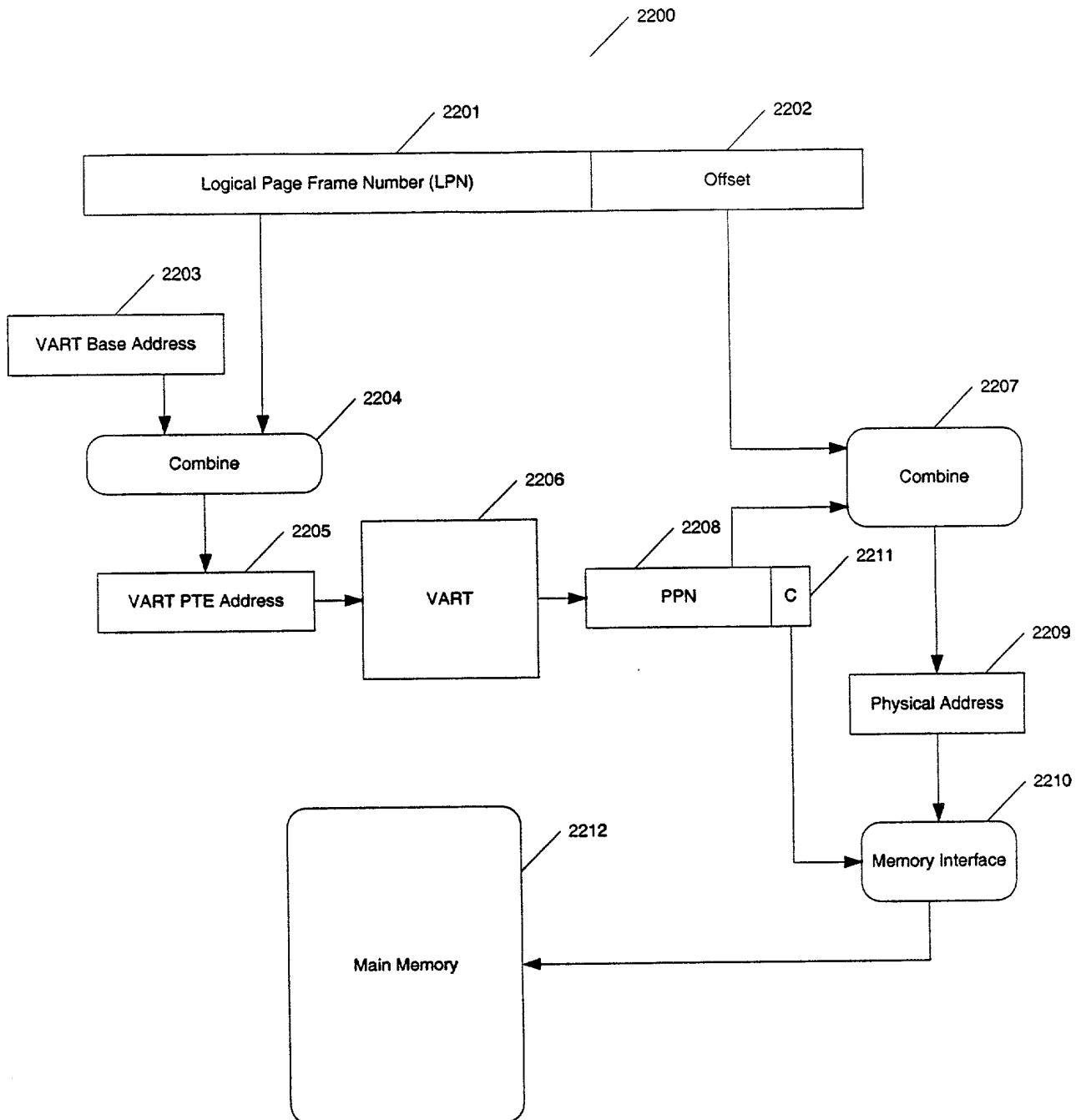


Figure 12

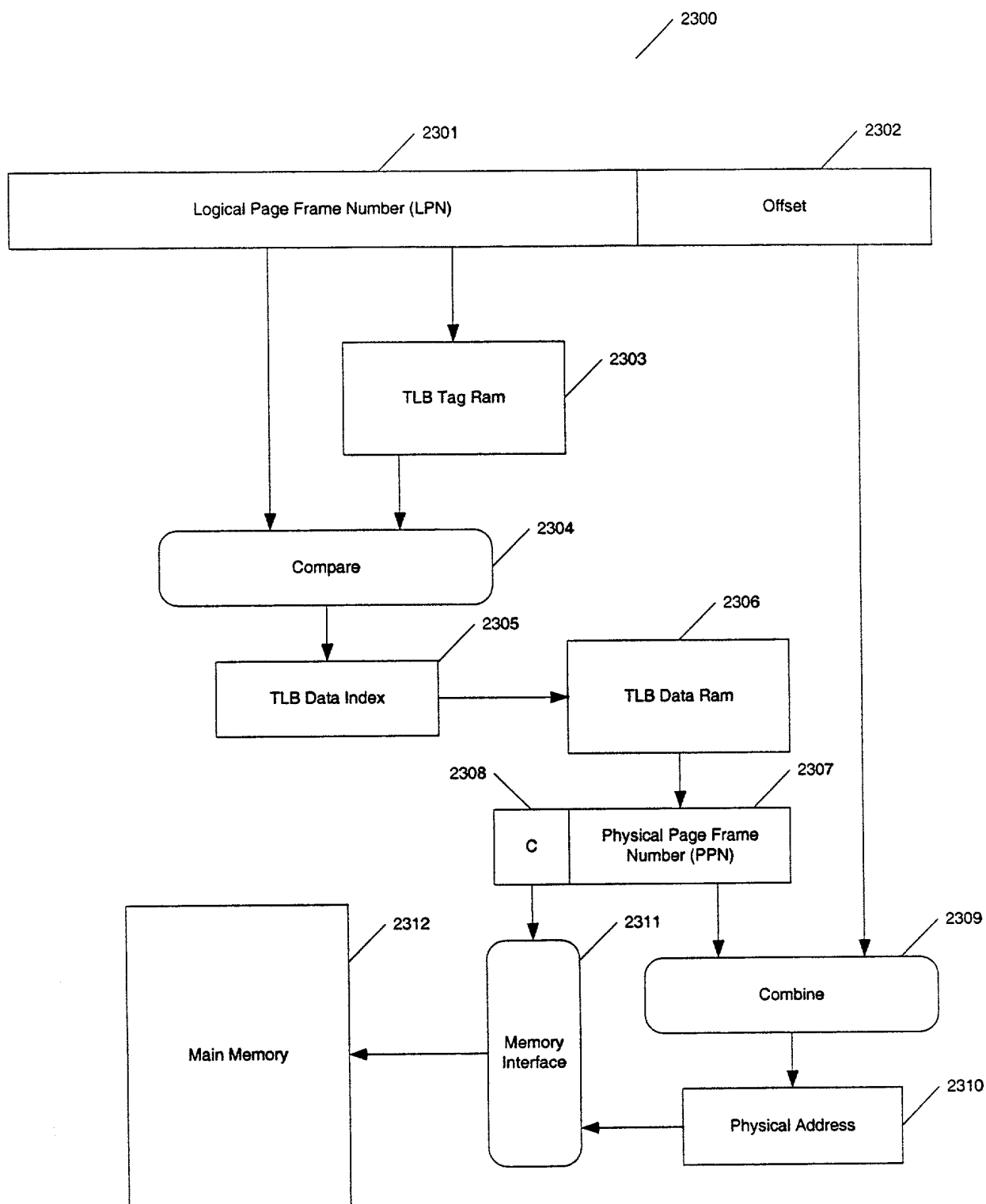


Figure 13

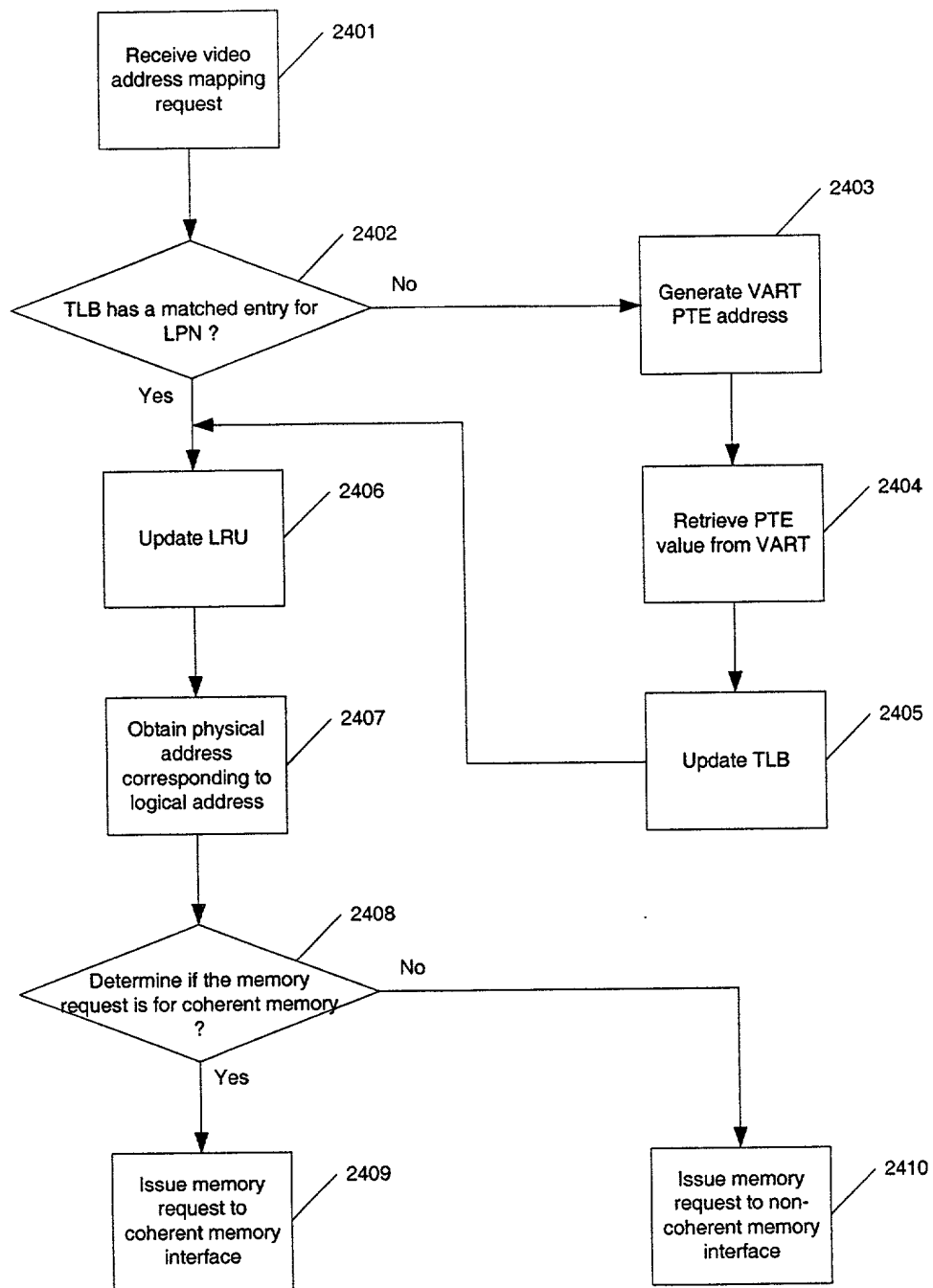


Figure 14

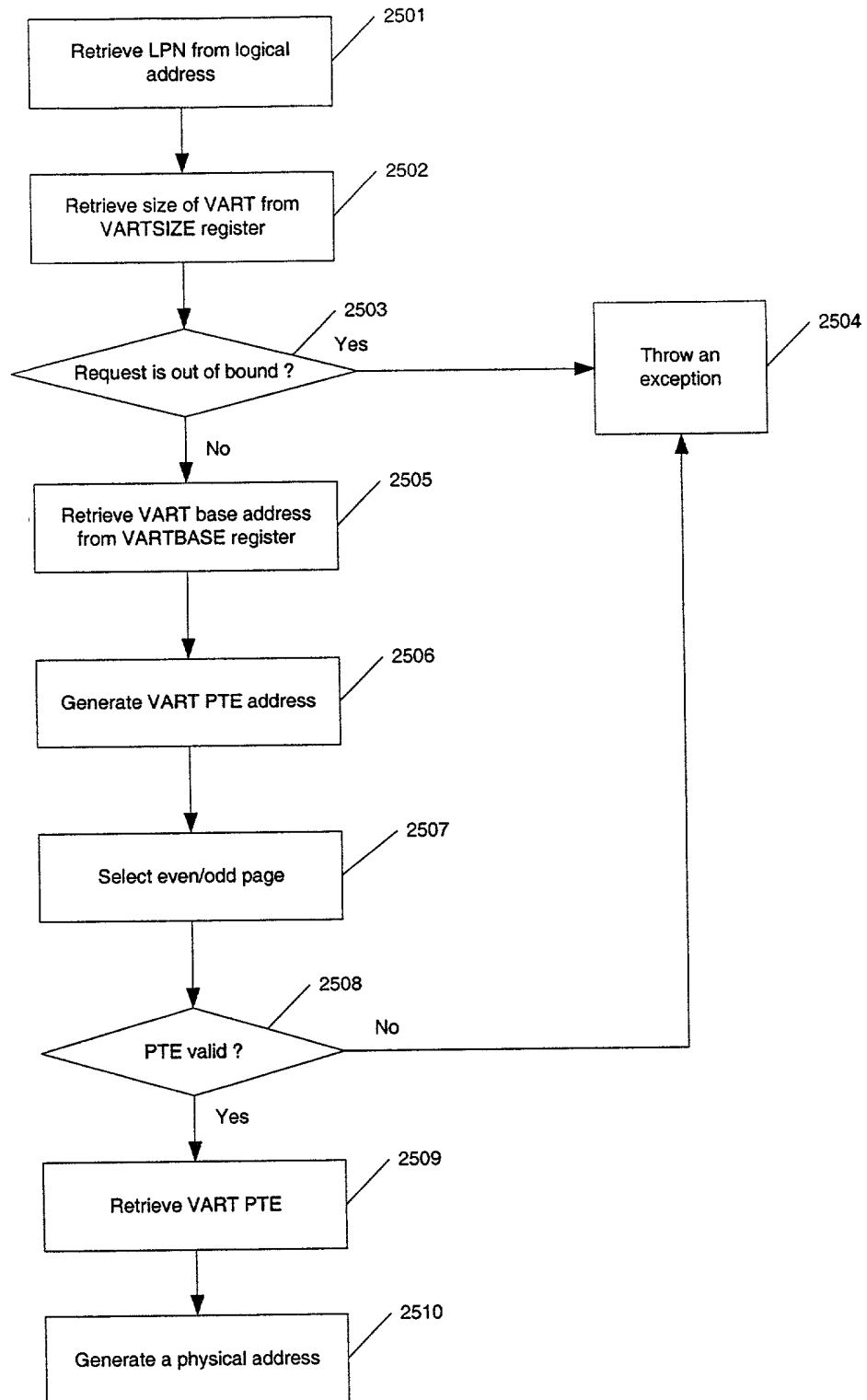


Figure 15

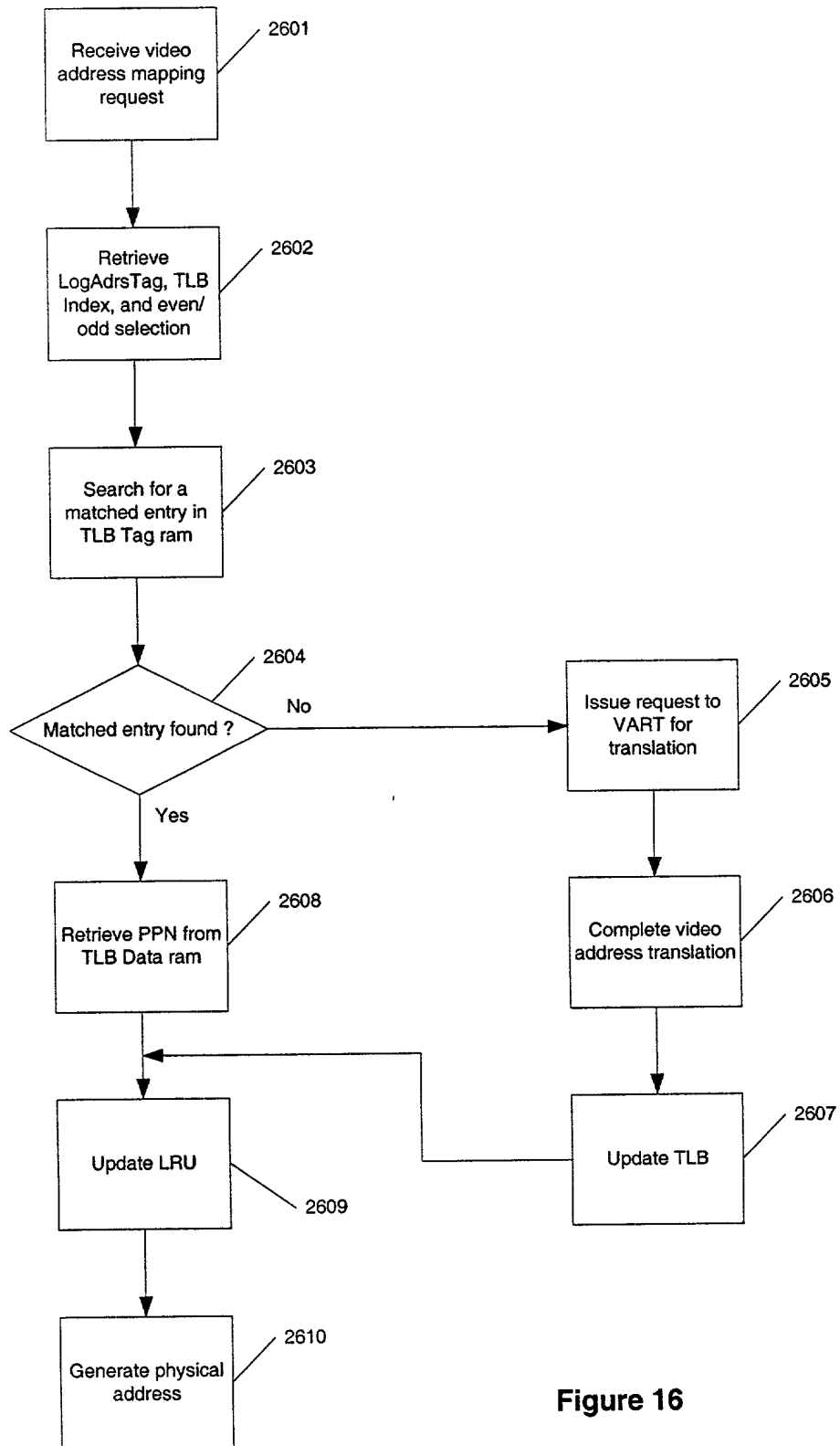


Figure 16

10038389 10038389

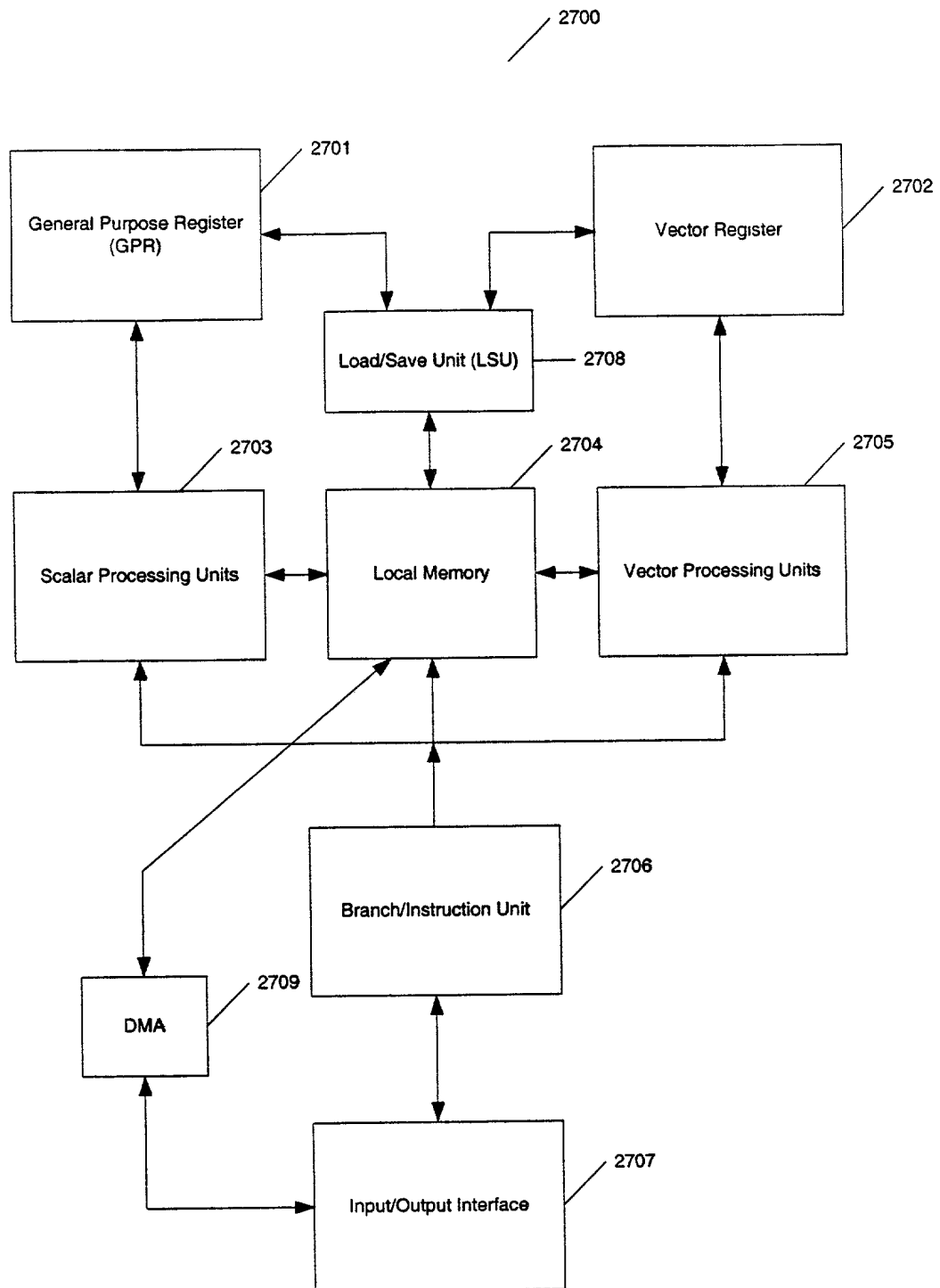


Figure 17

Figure 18

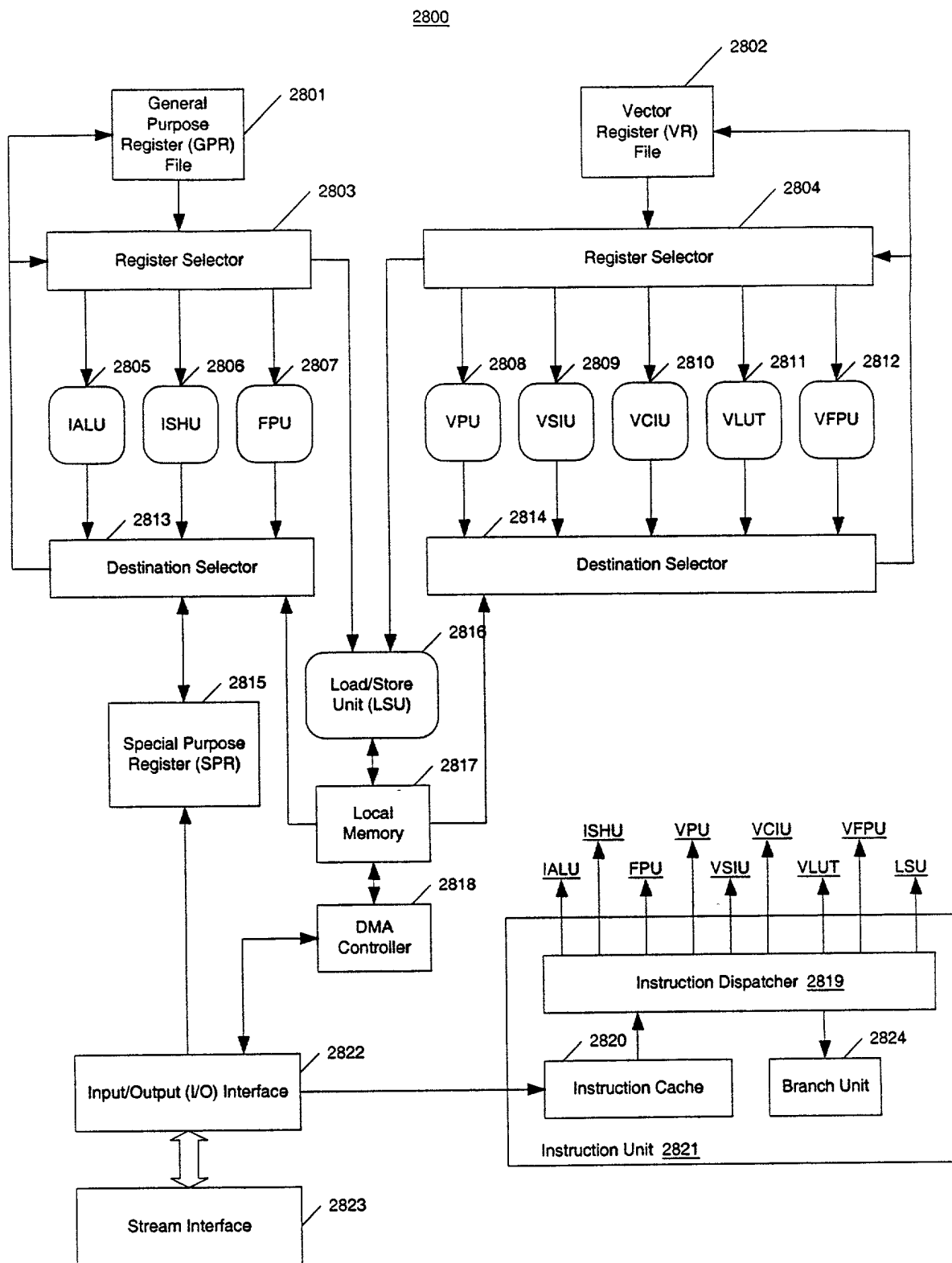


Figure 18

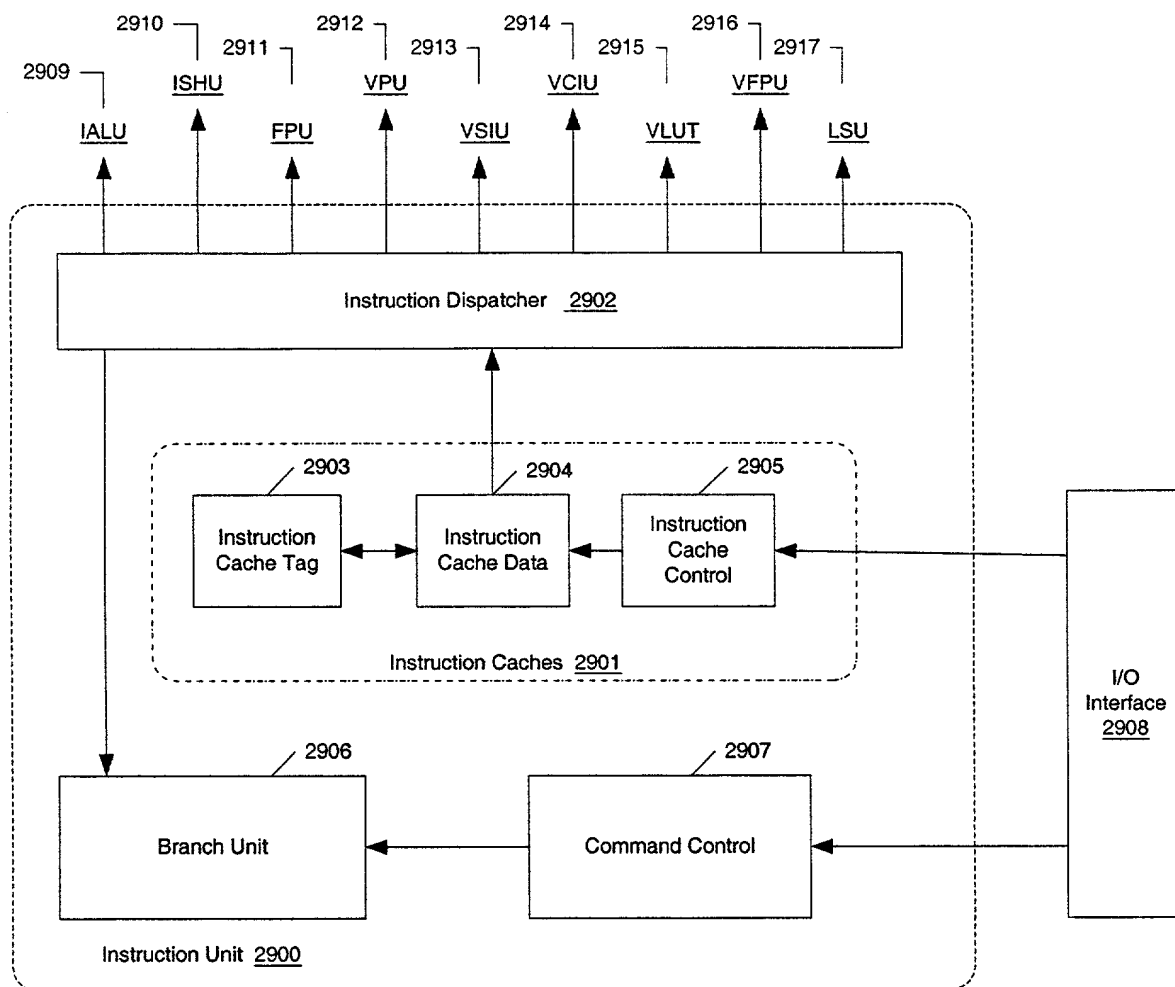


Figure 19A

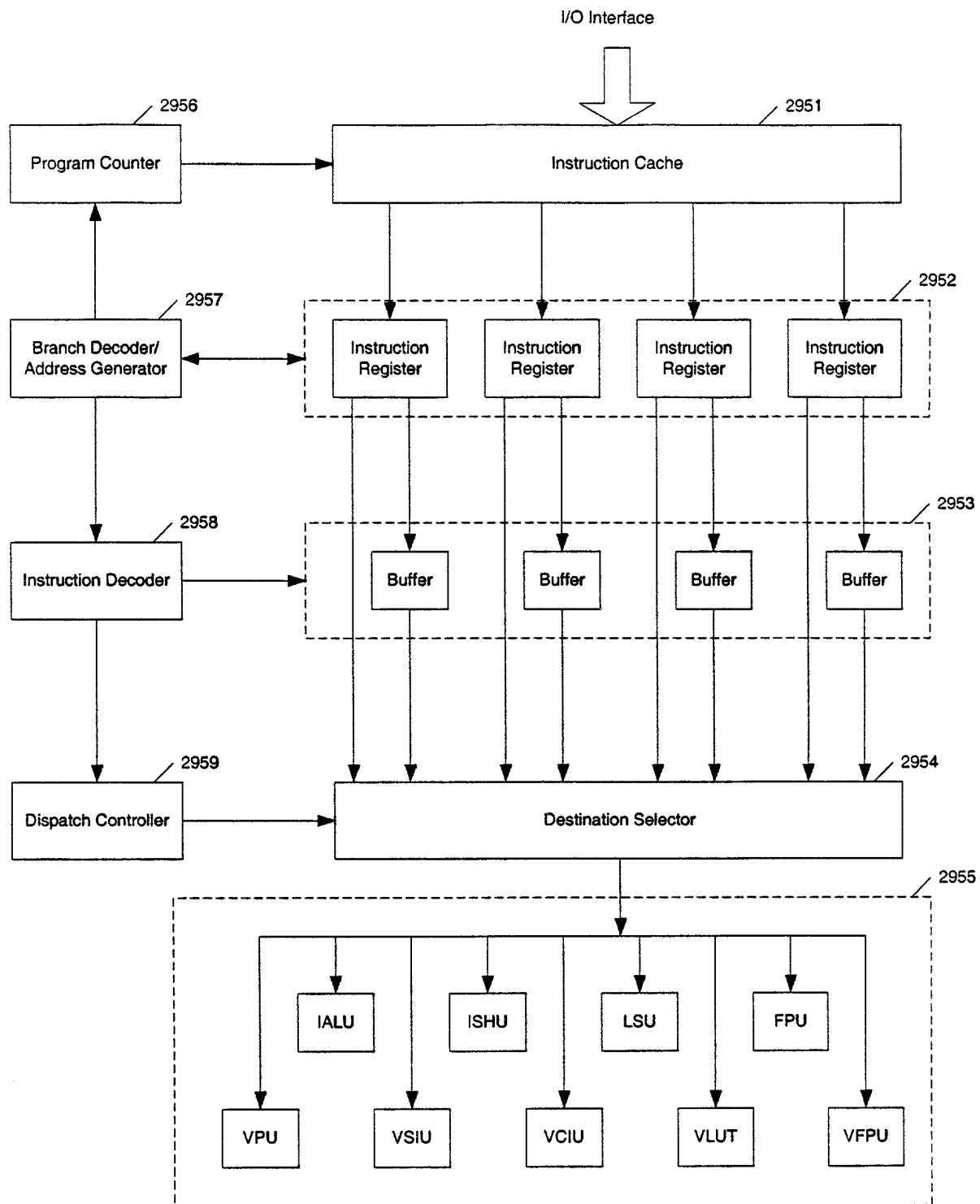


Figure 19B

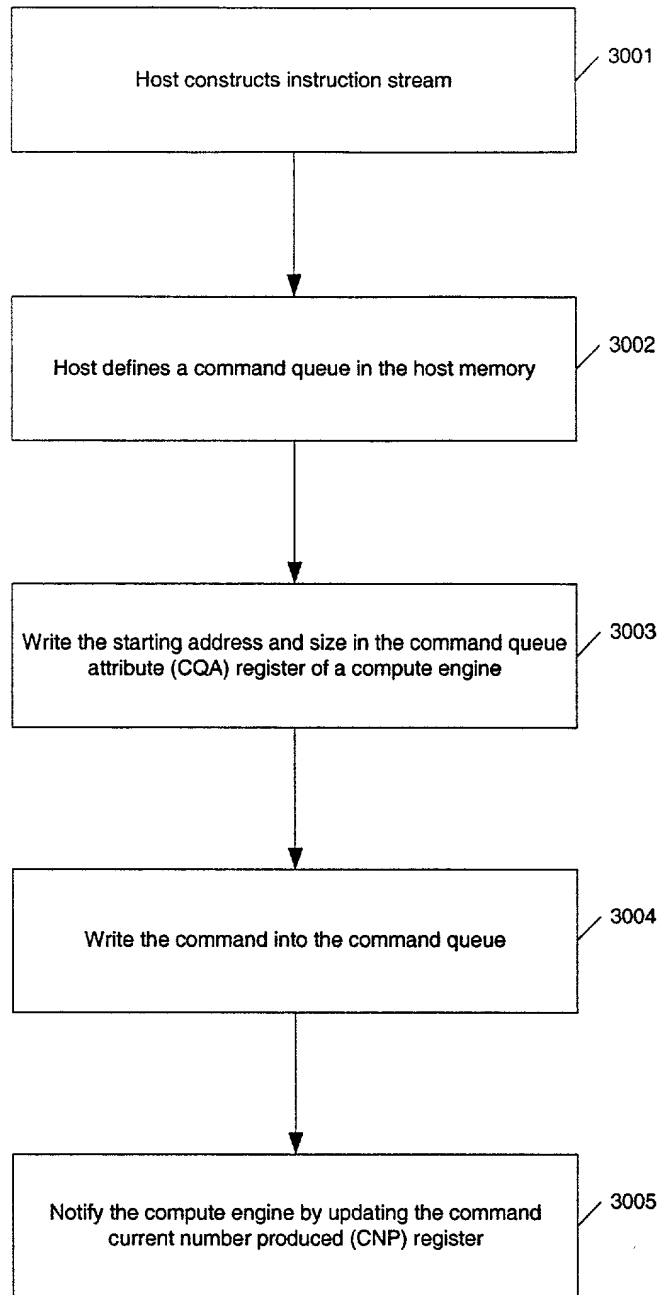


Figure 20A

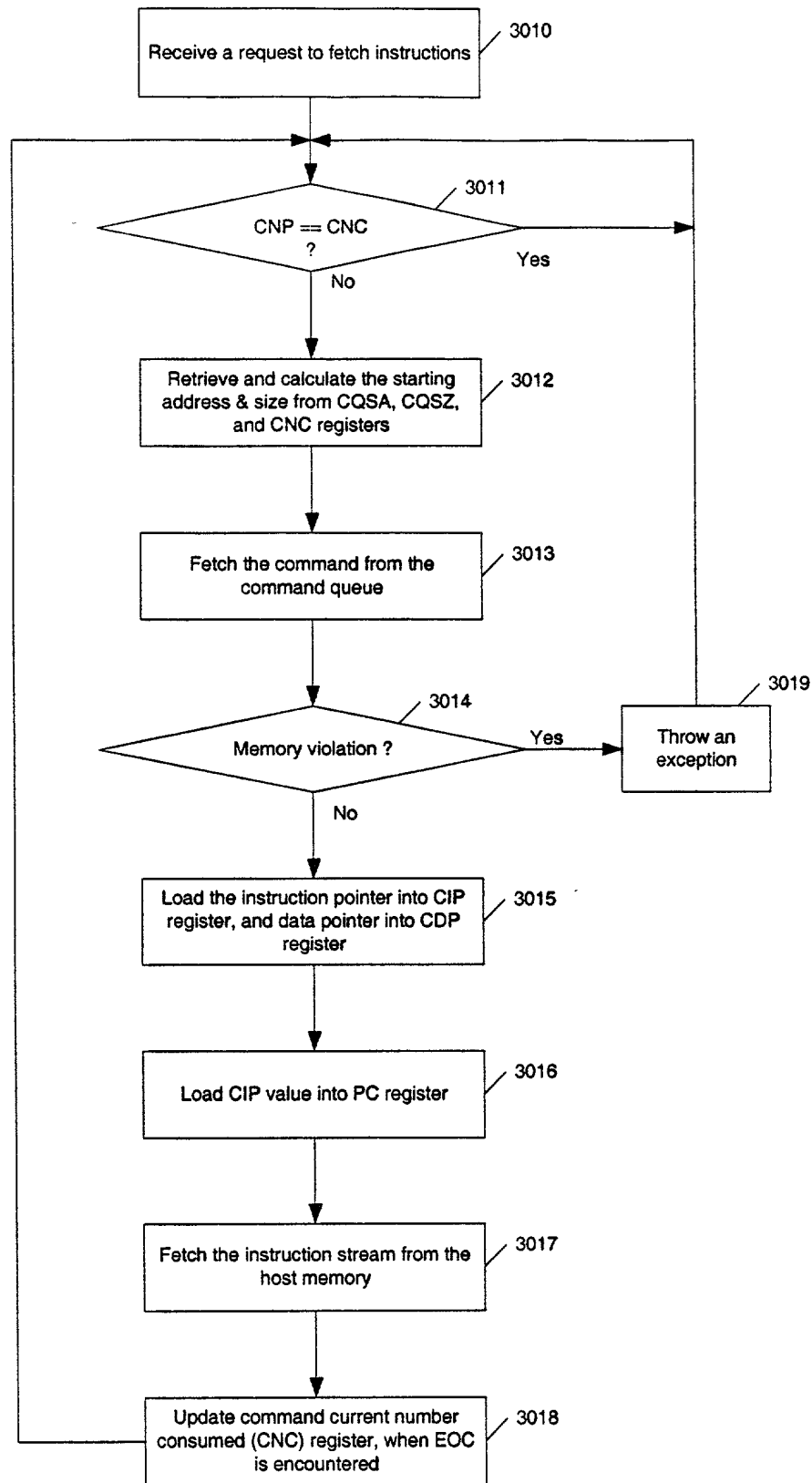


Figure 20B

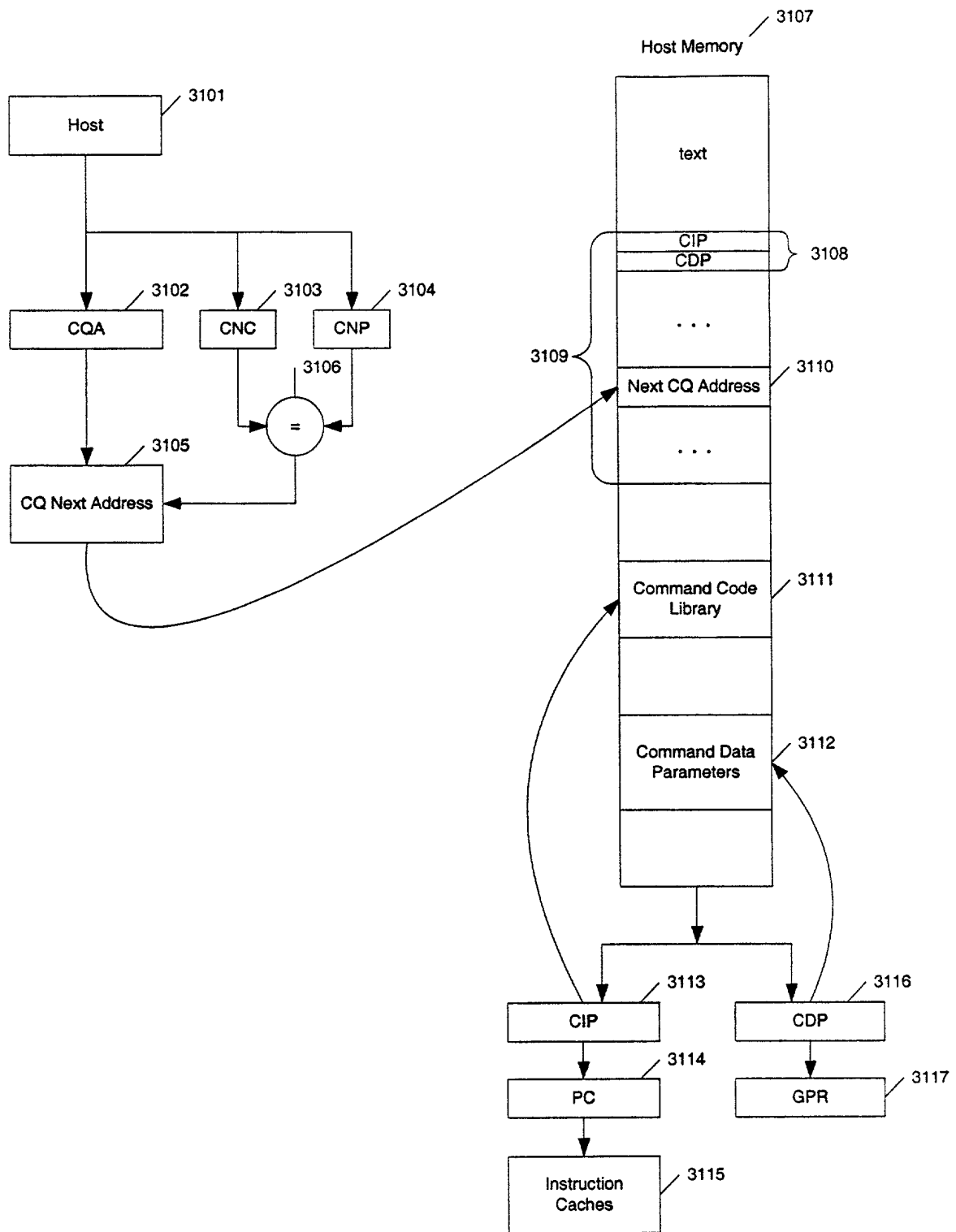
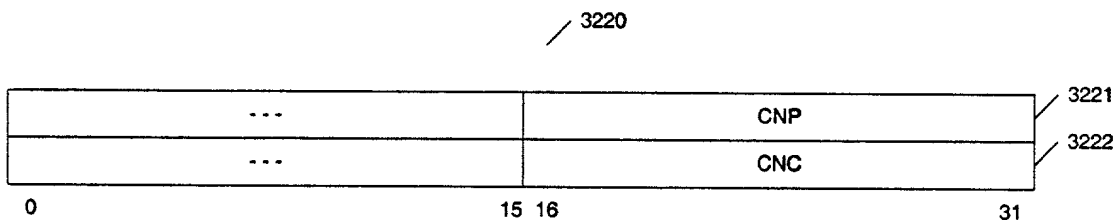
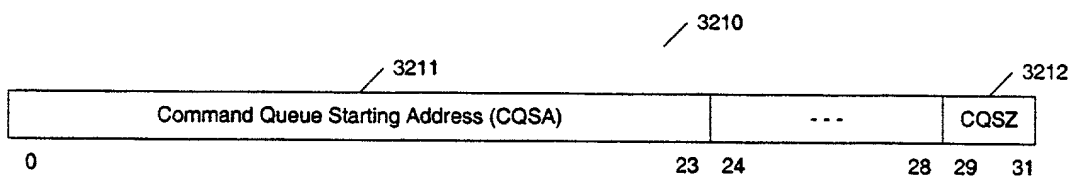
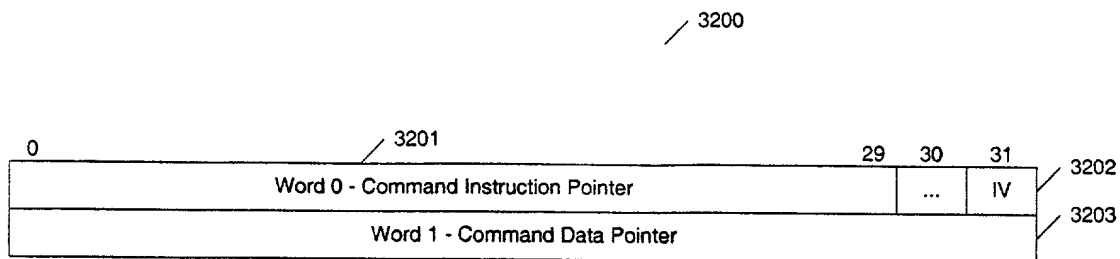


Figure 21



3230

| Command Queue Size (CQSZ) Code and Next Address Generation | | | | |
|--|--------------|-----------------------------------|-------------|-----|
| Code | Size (bytes) | Command Queue Next Address [0:31] | | |
| 000 | 256 | CQSA [0:23] | CNC [11:15] | 000 |
| 001 | 512 | CQSA [0:22] | CNC [10:15] | 000 |
| 010 | 1024 | CQSA [0:21] | CNC [9:15] | 000 |
| 011 | 2048 | CQSA [0:20] | CNC [8:15] | 000 |
| 100 | 4096 | CQSA [0:19] | CNC [7:15] | 000 |

0 19 20 21 22 23 24 28 29 31

3231 3232 3233

Figure 22

3300

| Priority Number | Functional Group Name | 3301 |
|-----------------|--|------|
| 0 | IALU - Integer Arithmetic/Logical Unit | |
| 1 | ISHU - Integer Shift Unit | |
| 2 | LSU - Load/Store Unit | |
| 3 | VPU - Vector Permute Unit | |
| 4 | VSIU - Vector Simple Integer Unit | |
| 5 | VCIU - Vector Complex Integer Unit | |
| 6 | VLUT - Vector Look-up Table Unit | |
| 7 | BRU - Branch Unit | 3302 |

Figure 23

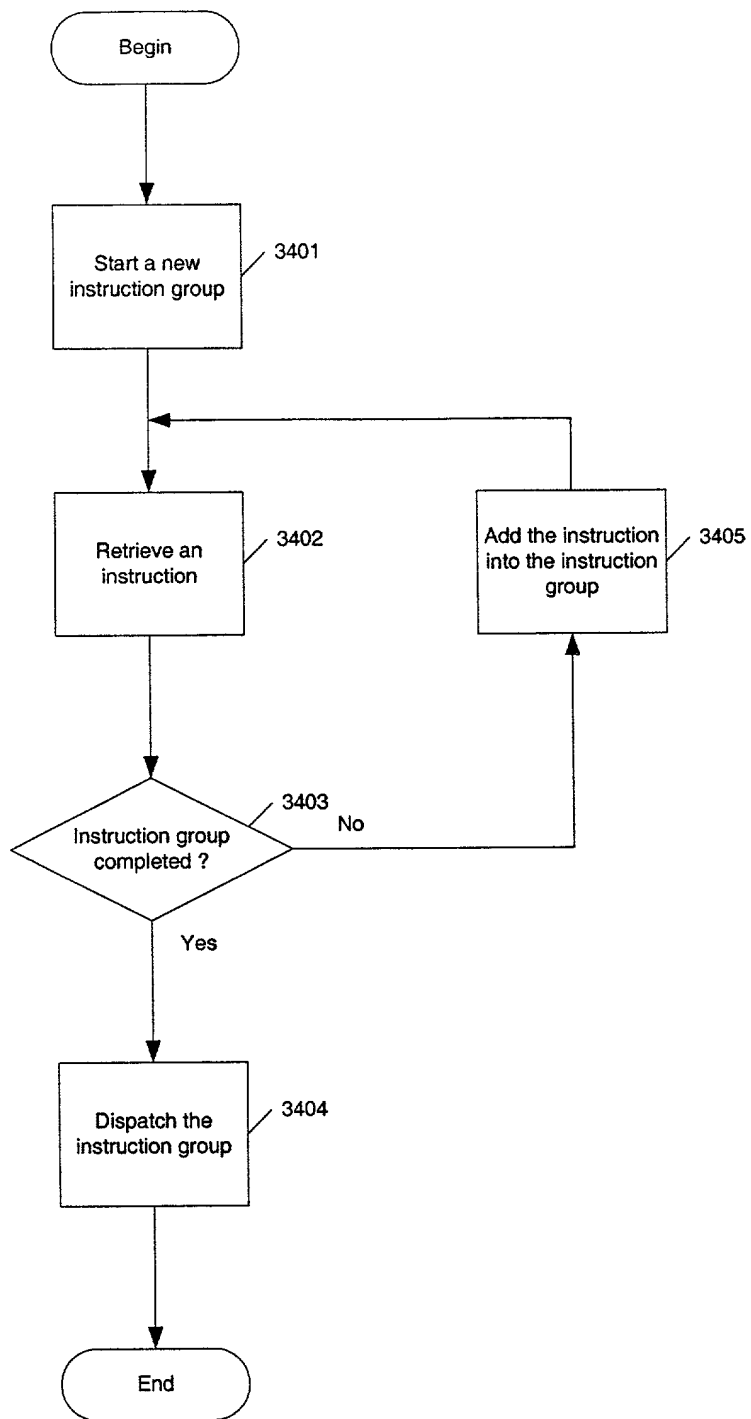


Figure 24

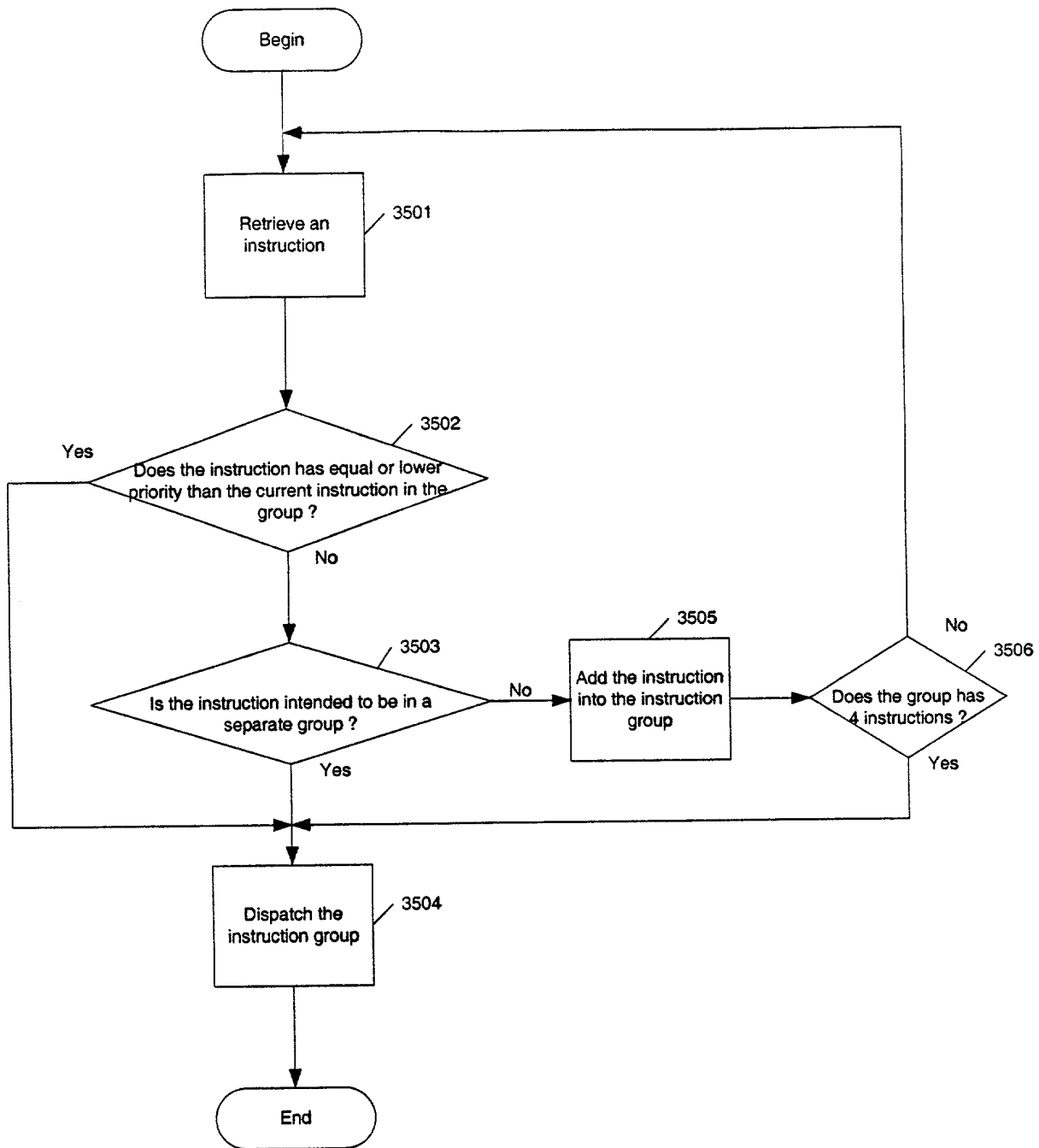


Figure 25

| Functional Unit | Latency | Dispatch Rate |
|--------------------------------|---------|---------------|
| IALU - not multiply or divide | 2 | 1 |
| IALU - multiply | 19 | 19 |
| IALU - divide | 35 | 35 |
| ISHU | 2 | 1 |
| LSU - non-DMA address update | 2 | 1 |
| LSU - non-DMA load data update | 3 | 1 |
| LSU - non-DMA store | 1 | 1 |
| LSU - DMA instructions | 1 | 1 |
| VPU | 2 | 1 |
| VSIU | 2 | 1 |
| VCIU | 6 | 1 |
| VLUT - reads, vwid | 4 | 1 |
| VLUT - writes | 1 | 1 |
| Branch instruction | 1 | 1 |

Figure 26

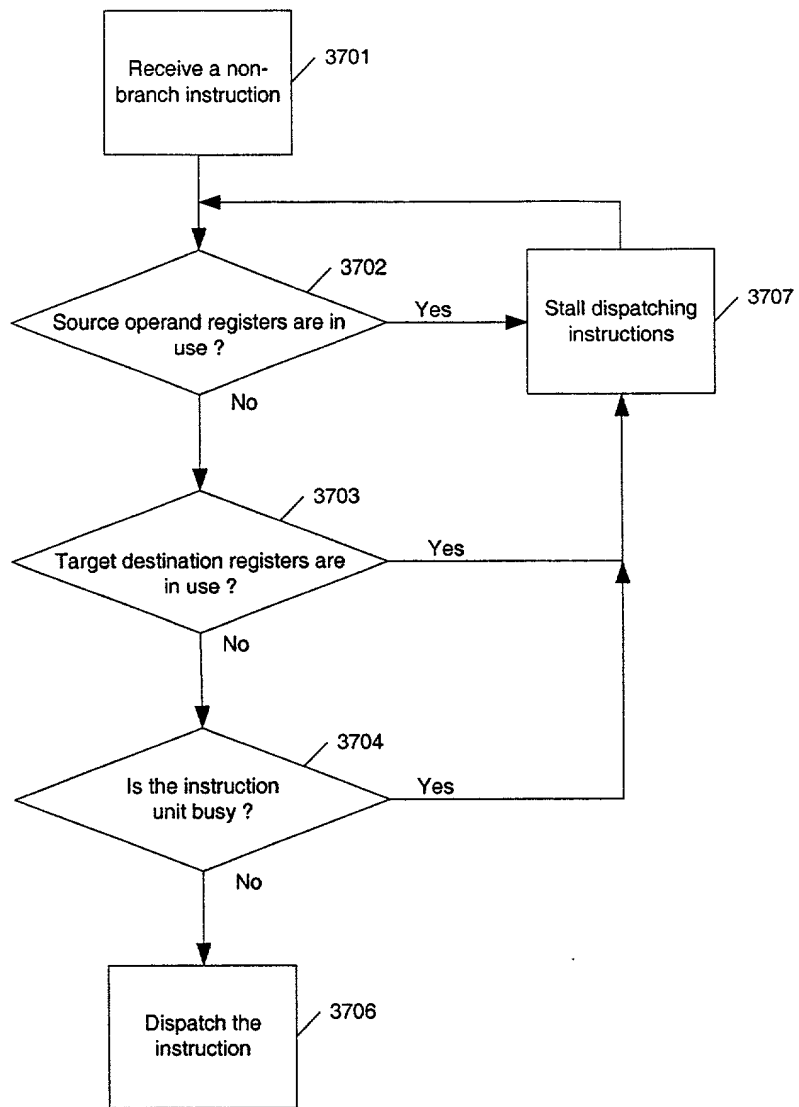


Figure 27

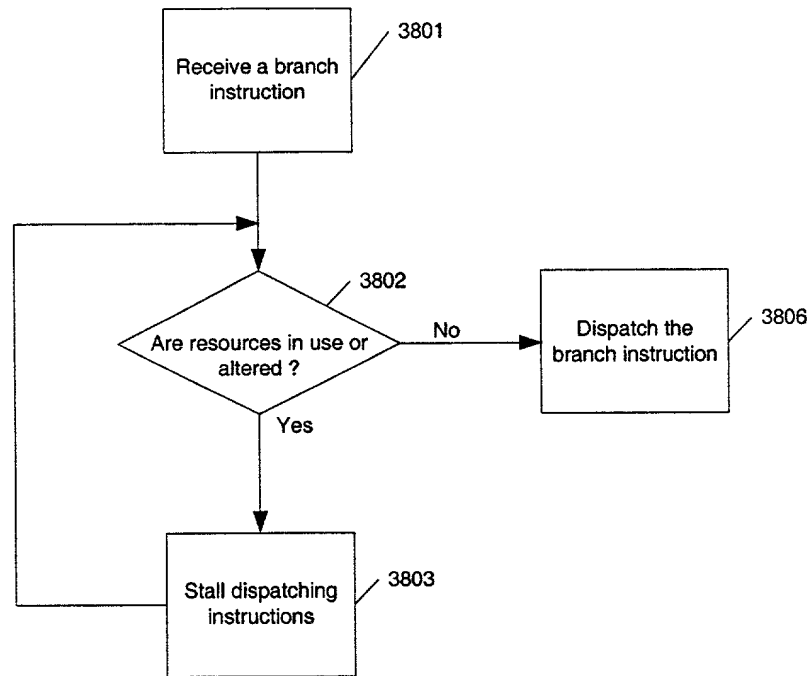
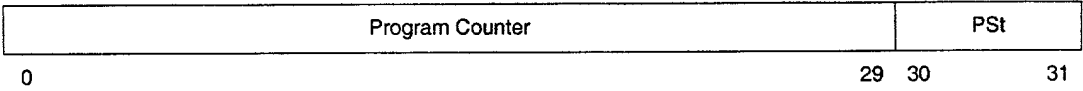


Figure 28

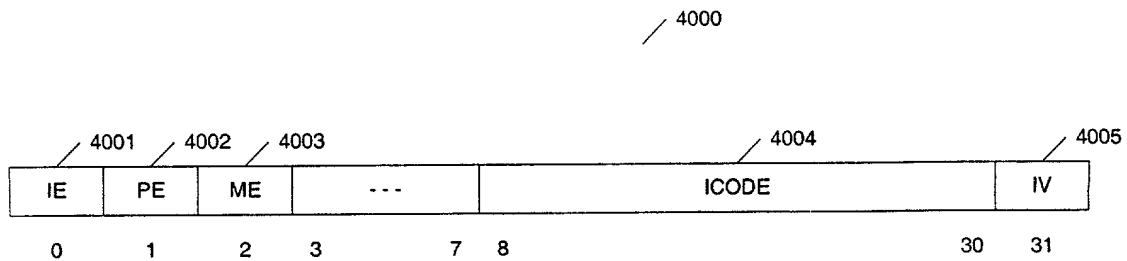
3900



3901

| Pst | Name | Description |
|-----|-------|--|
| 00 | Idle | CQ counters are equal and no current command executing. Program counter is invalid. |
| 01 | Run | Command was executing. Program counter points to next instruction that would have been executed. |
| 10 | IWait | Command was executing, but instruction fetching has stopped due to a previous exception. Program counter points to the next instruction that would have been executed. |
| 11 | CWait | Command was not executing due to an exception in fetching the command. Program counter is invalid. |

Figure 29



| Name | Descriptions |
|-------|--|
| IE | Illegal Opcode Exception. Occurs whenever an illegal Opcode is fetched for execution. Cleared when read by the host. |
| PE | Program Counter Exception. Occurs whenever the host does a read program counter with exception. Cleared with read by host. |
| ME | Memory Access Exception. Occurs whenever a memory operation results in a memory access exception. Cleared when read by the host. |
| ICODE | Interrupt Code. Can be read and written by a compute engine or the host. |
| IV | Interrupt Valid. Set and read by the compute engine to indicate and interrupt to the host. Read and cleared by the host. |

Figure 30

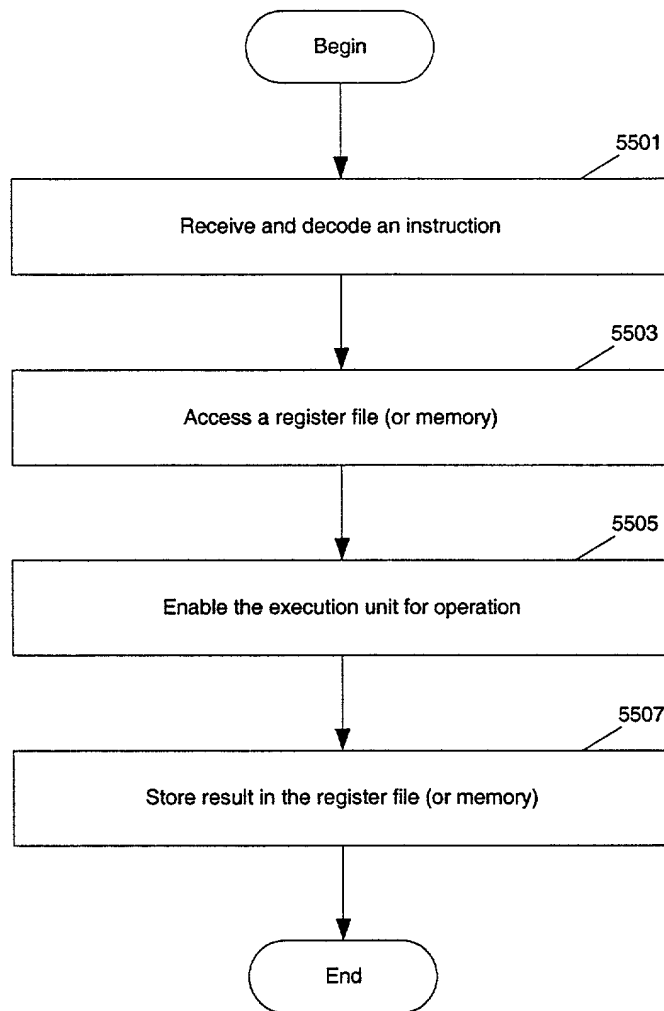


Fig. 31

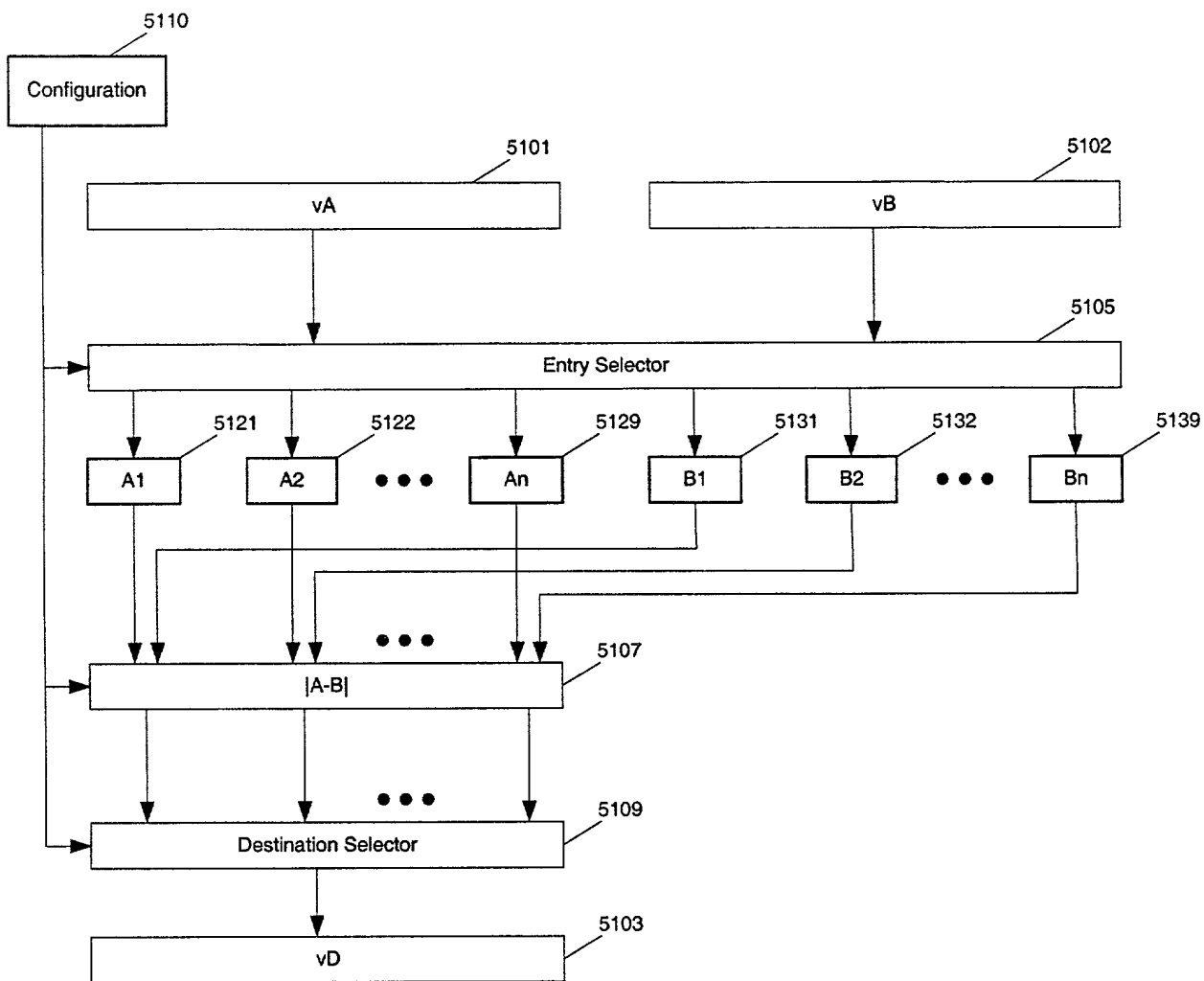


Fig. 32

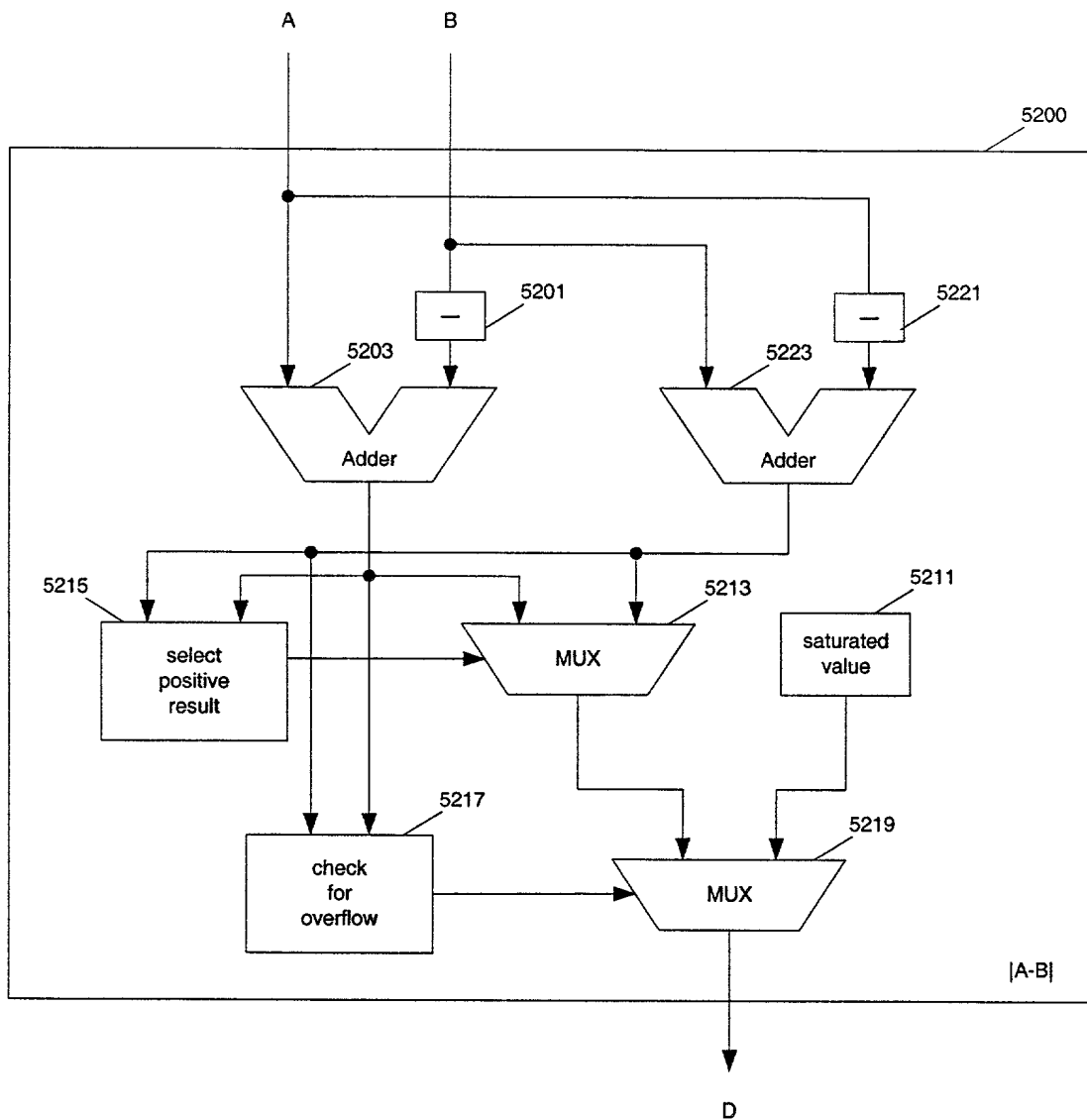


Fig. 33

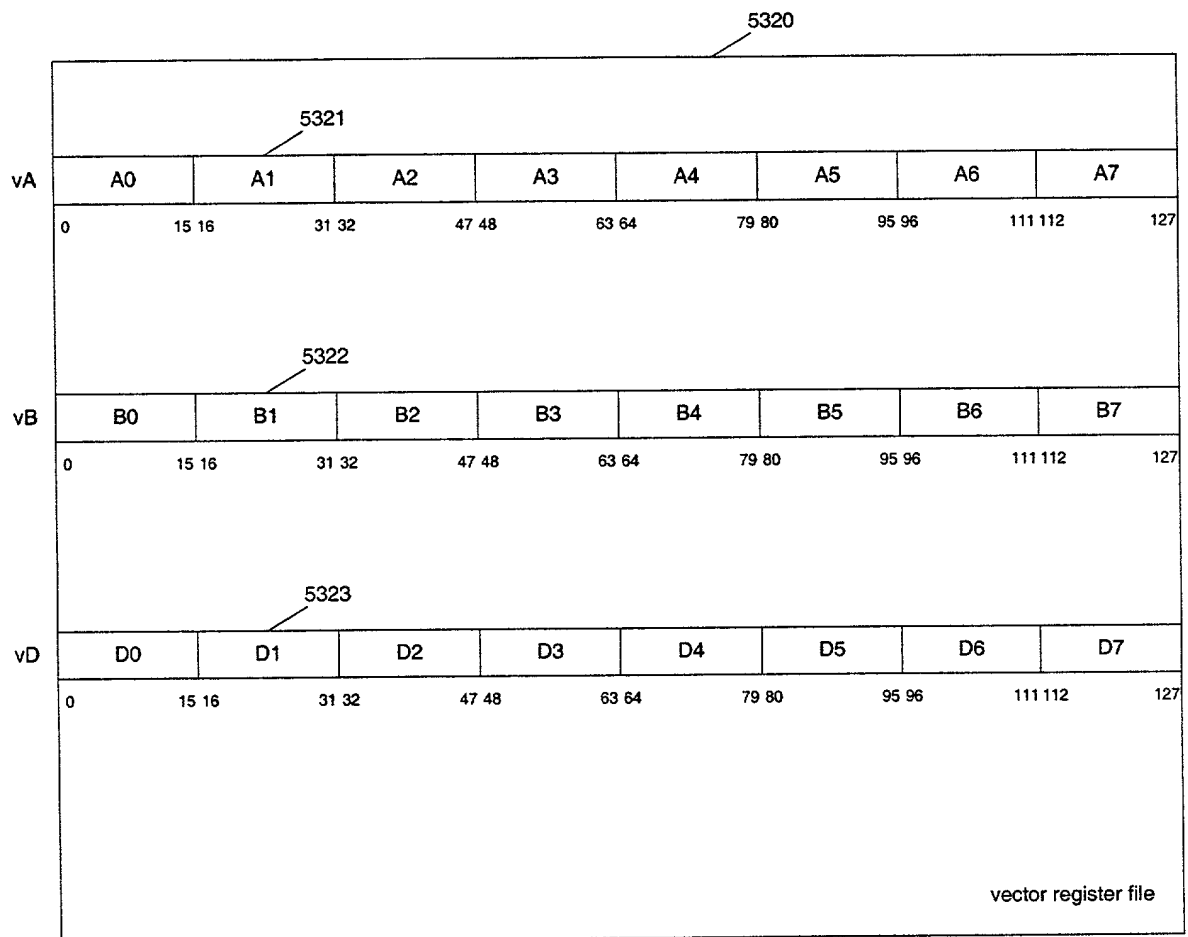
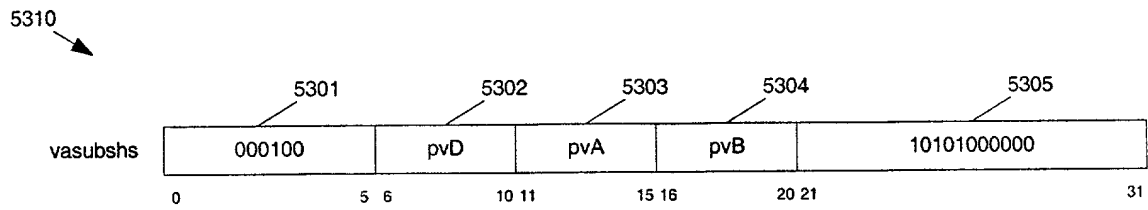


Fig. 34

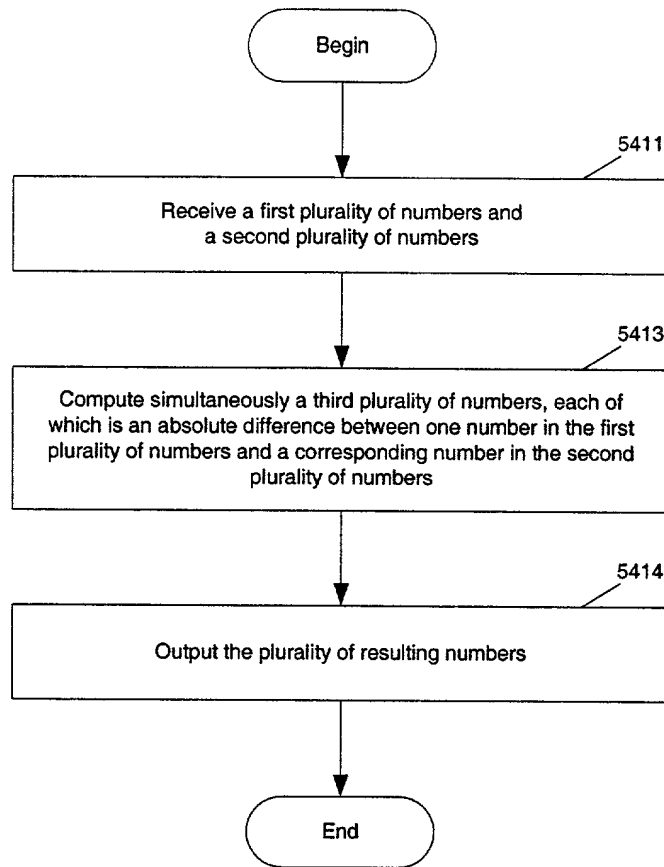


Fig. 35

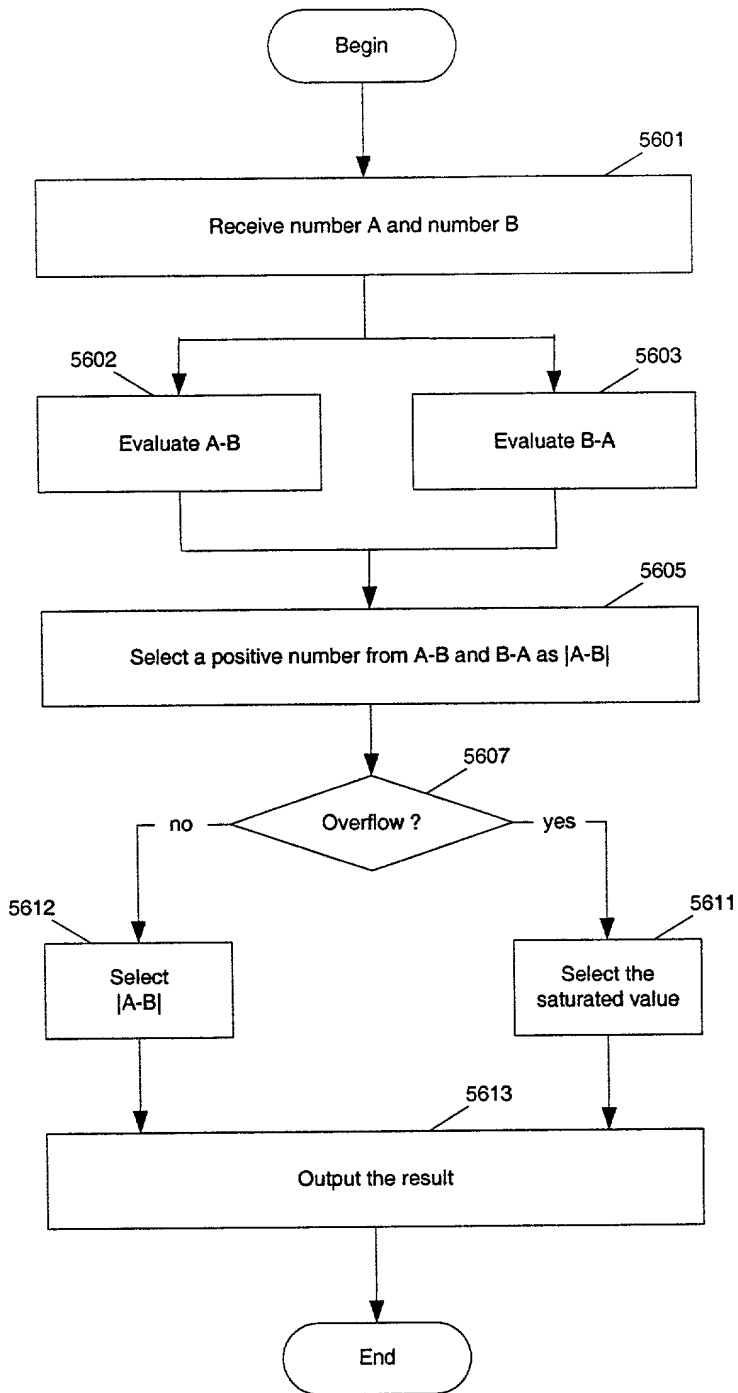


Fig. 36

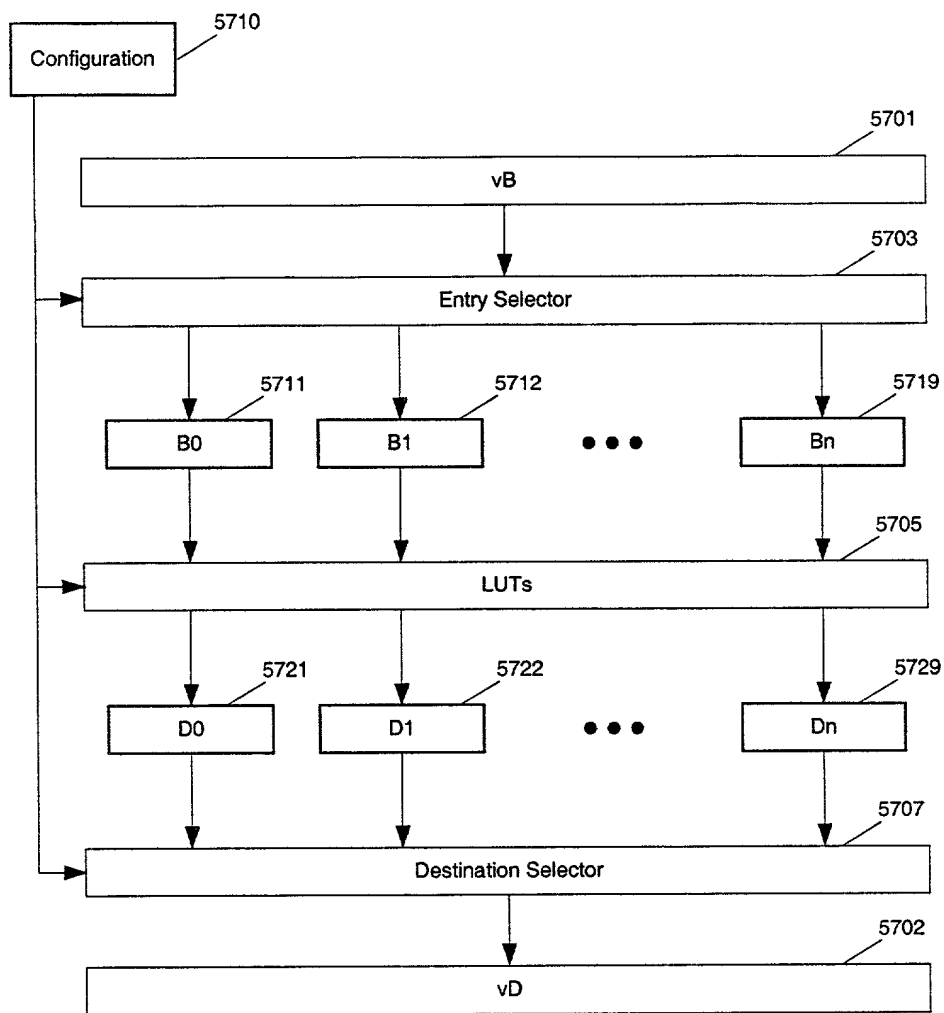


Fig. 37

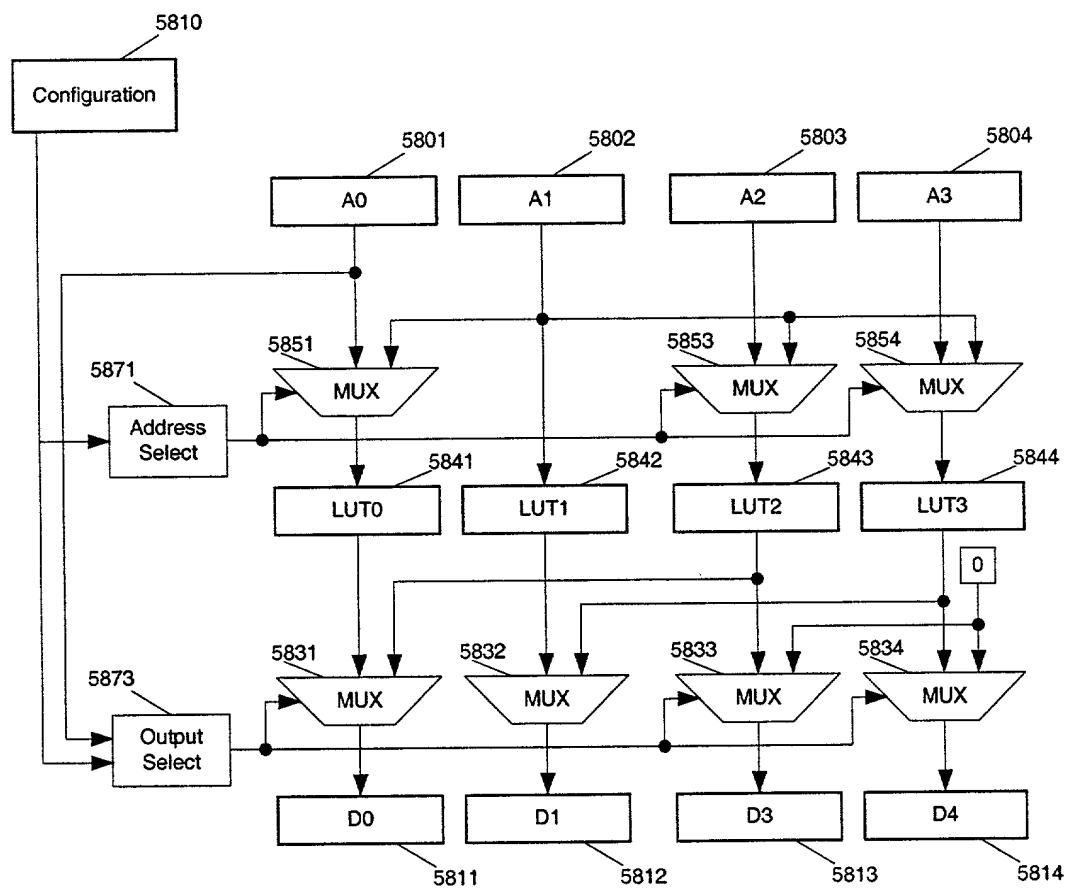


Fig. 38

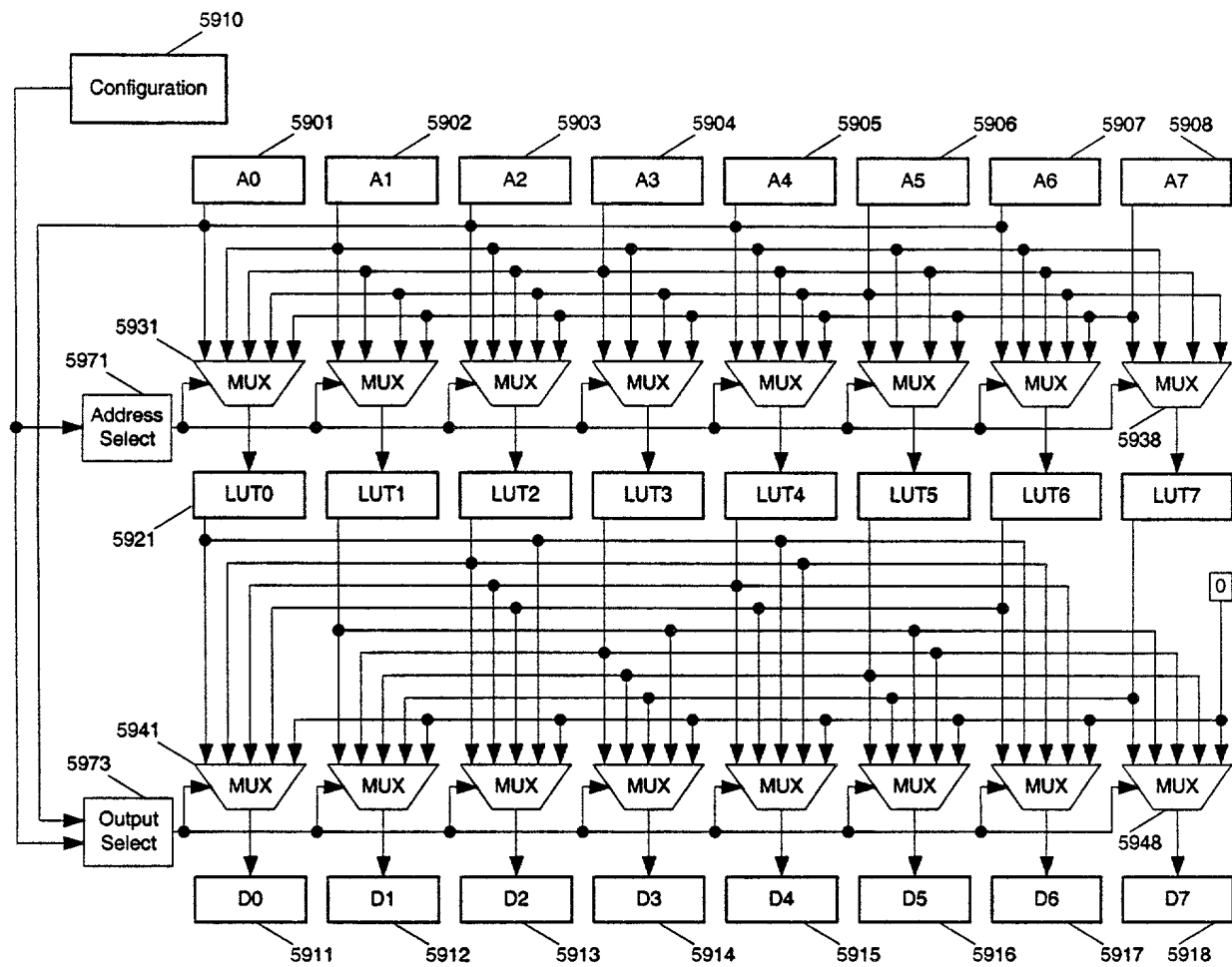


Fig. 39

6010

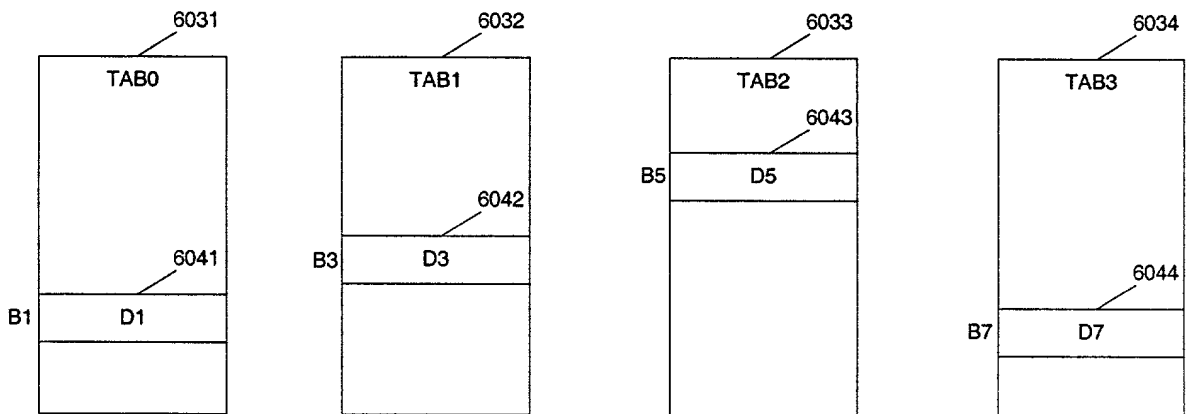
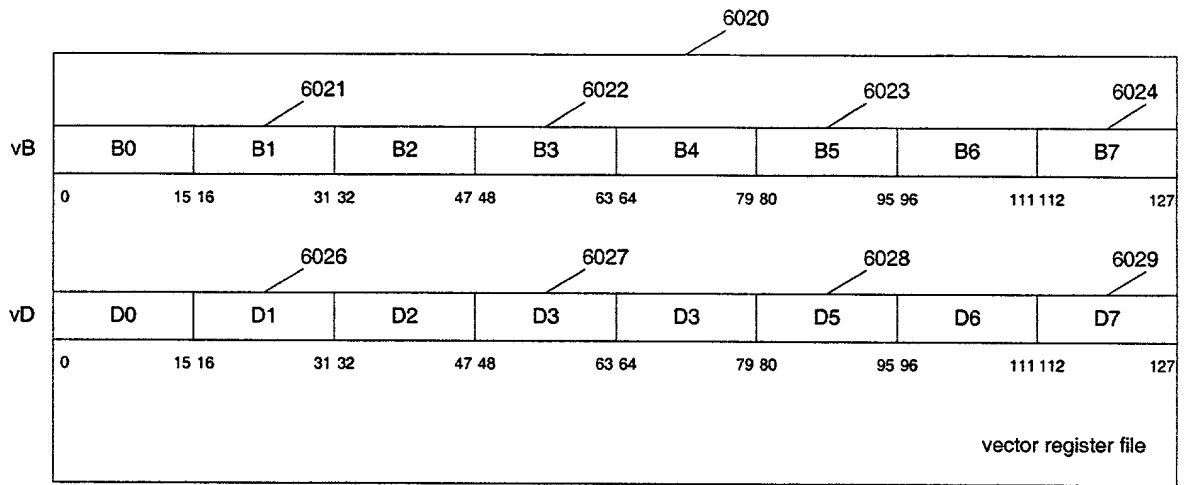
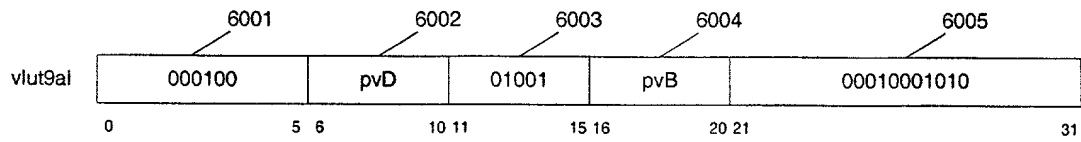


Fig. 40

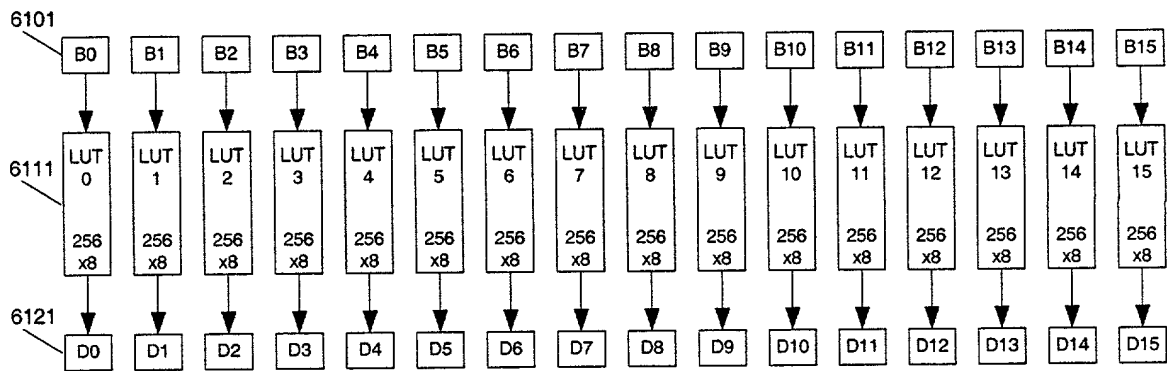


Fig. 41

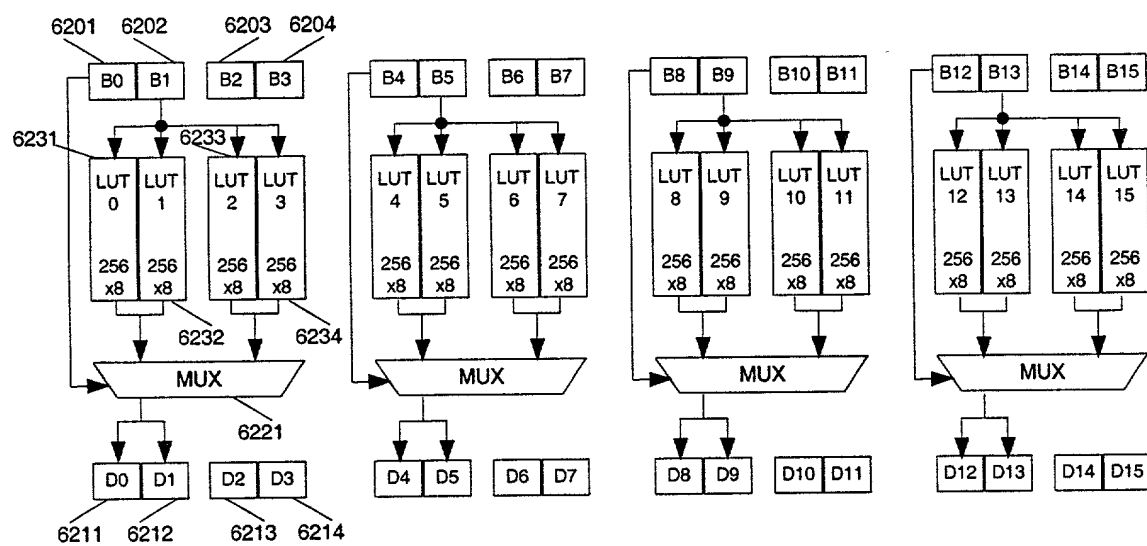


Fig. 42

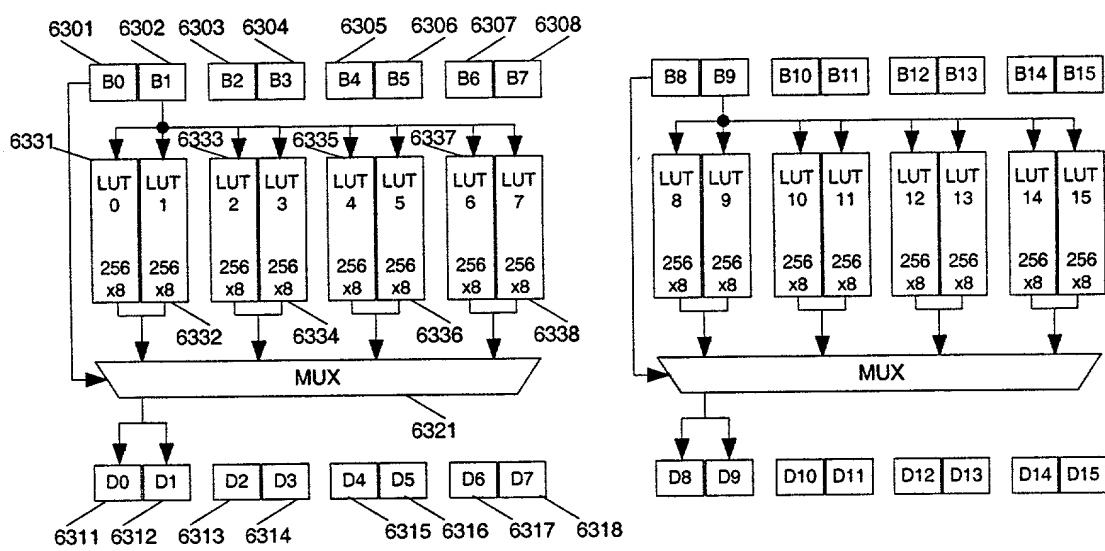


Fig. 43

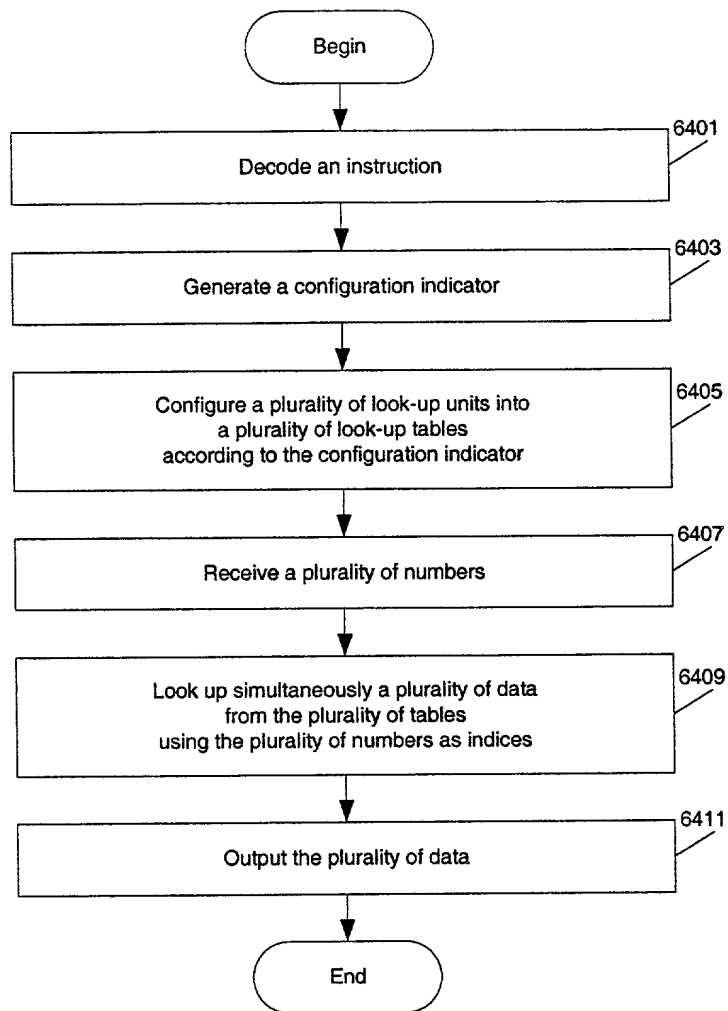


Fig. 44

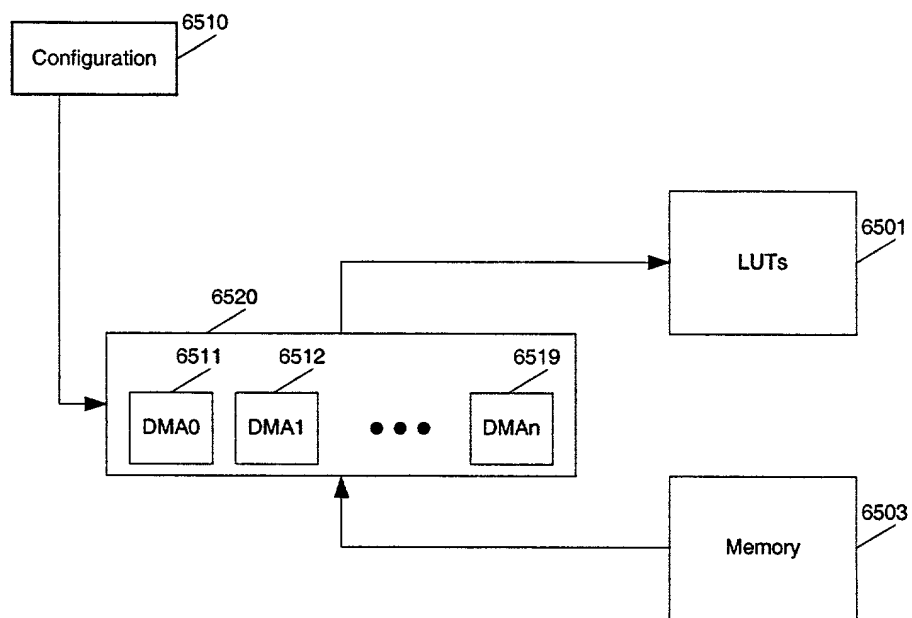


Fig. 45

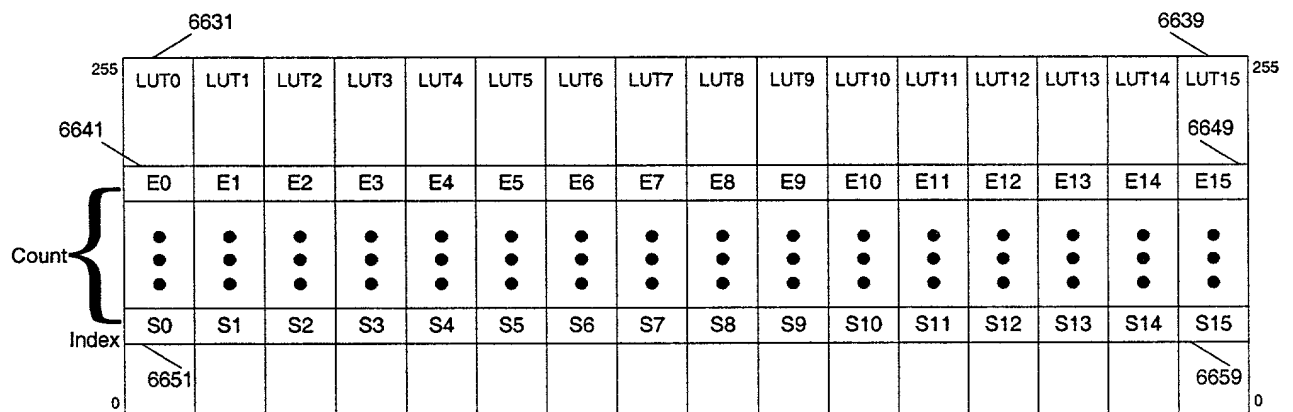
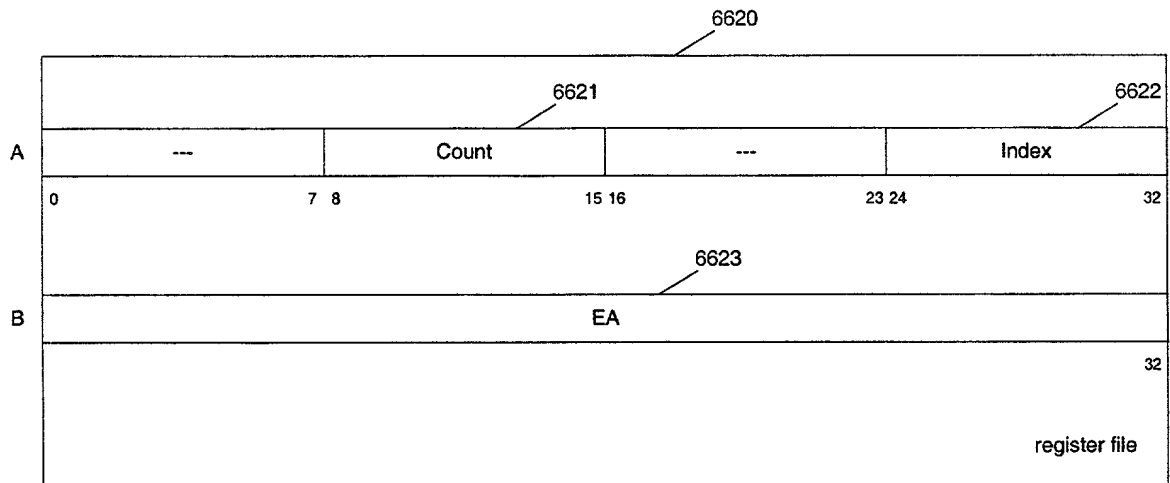
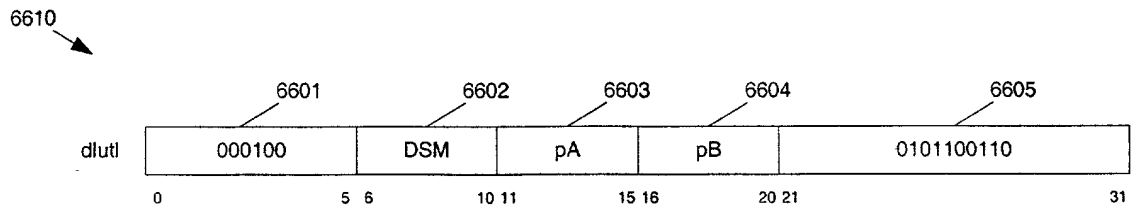


Fig. 46

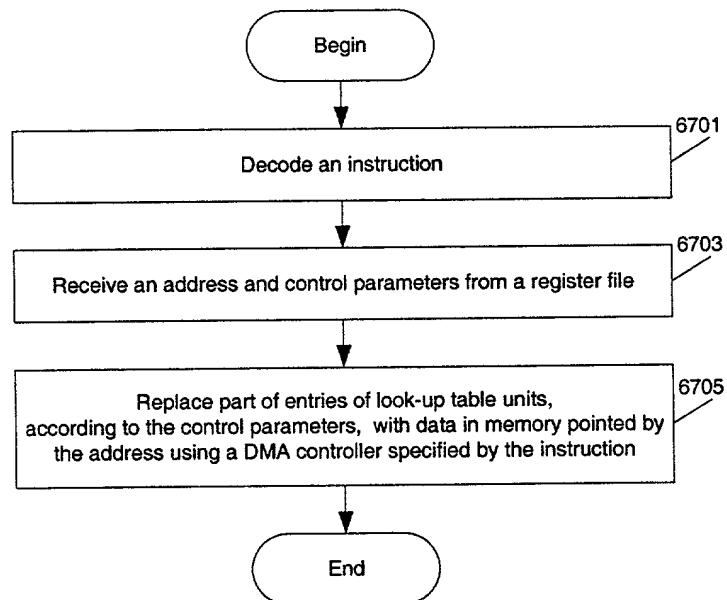


Fig. 47

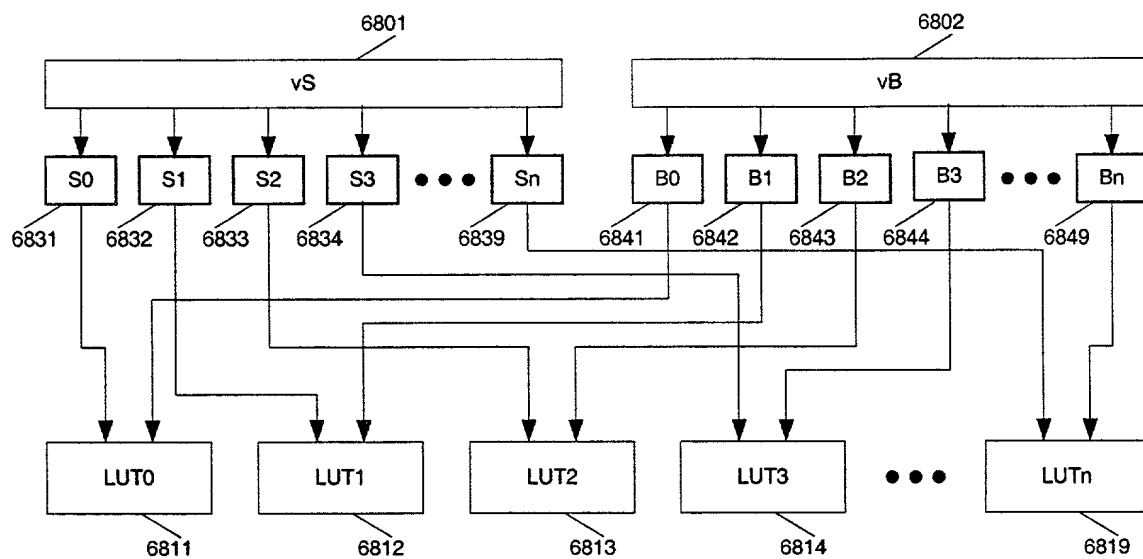


Fig. 48

6910

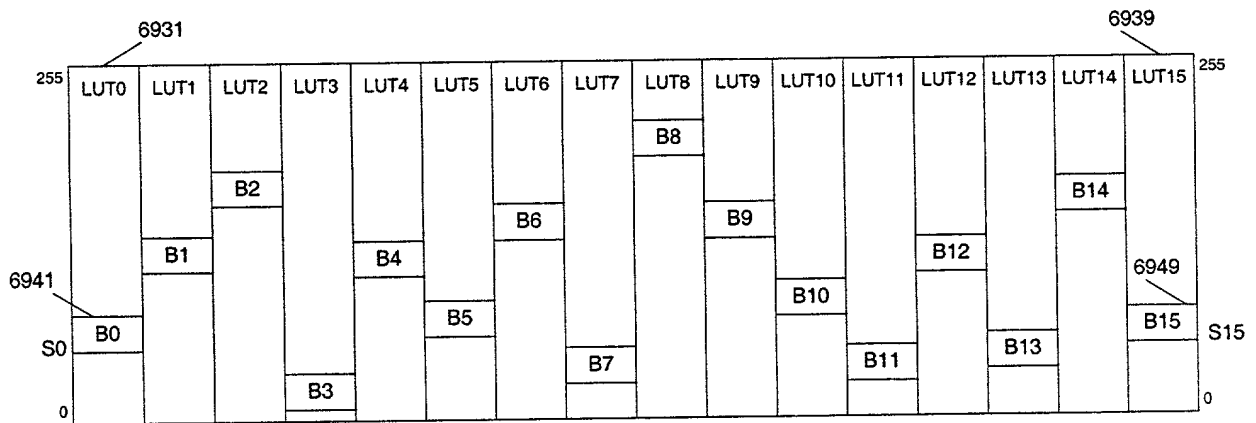
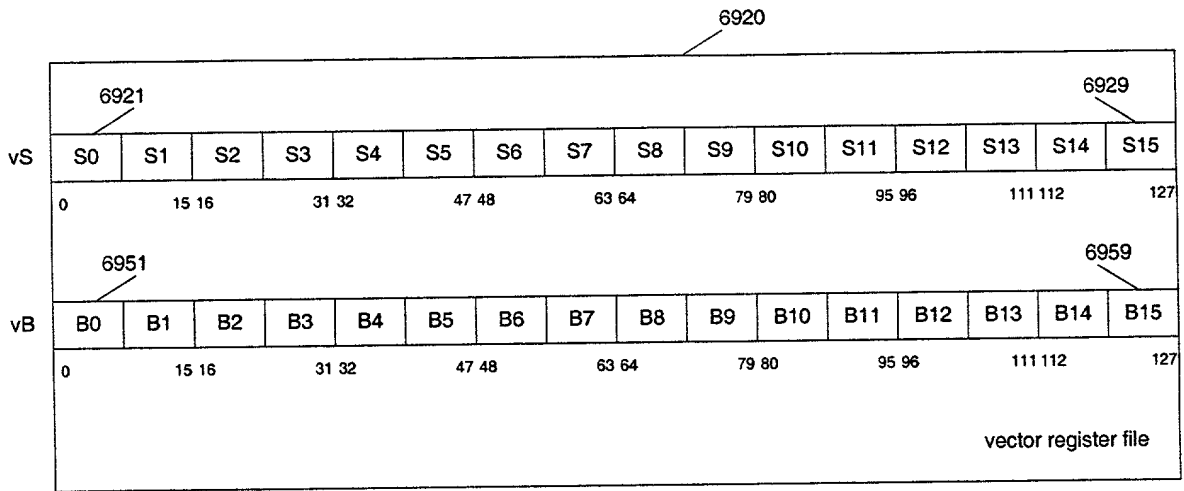
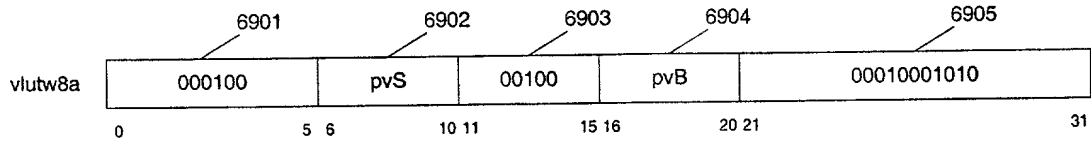


Fig. 49

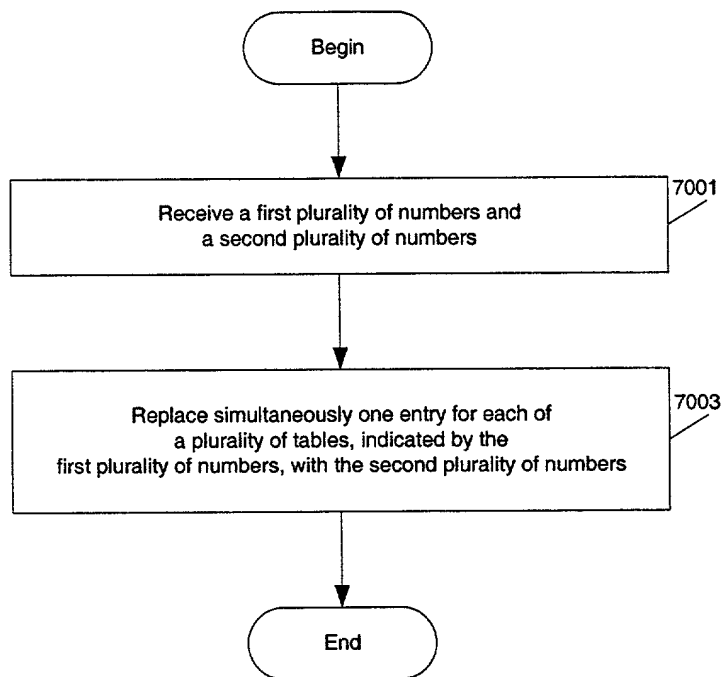


Fig. 50

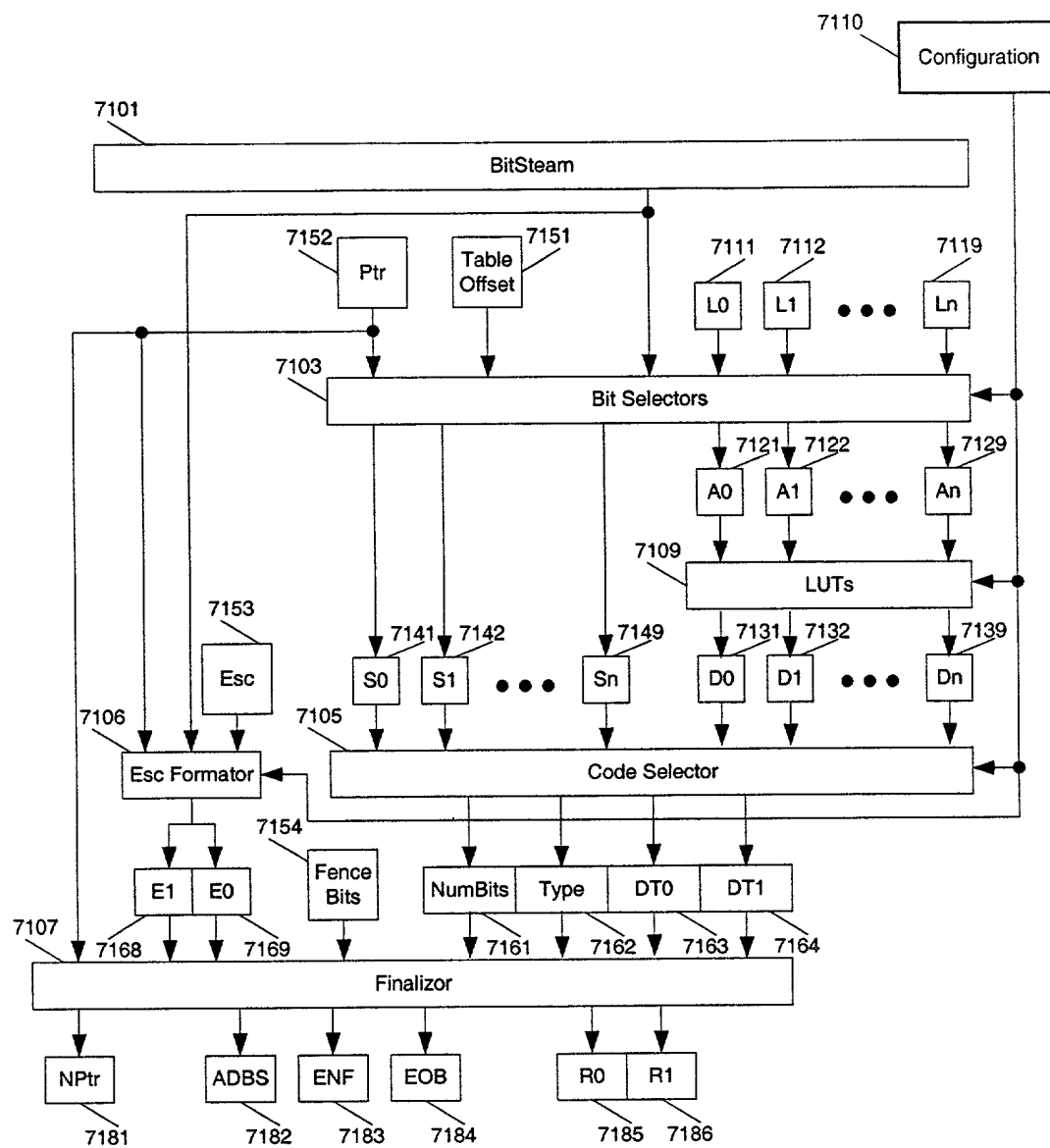


Fig. 51

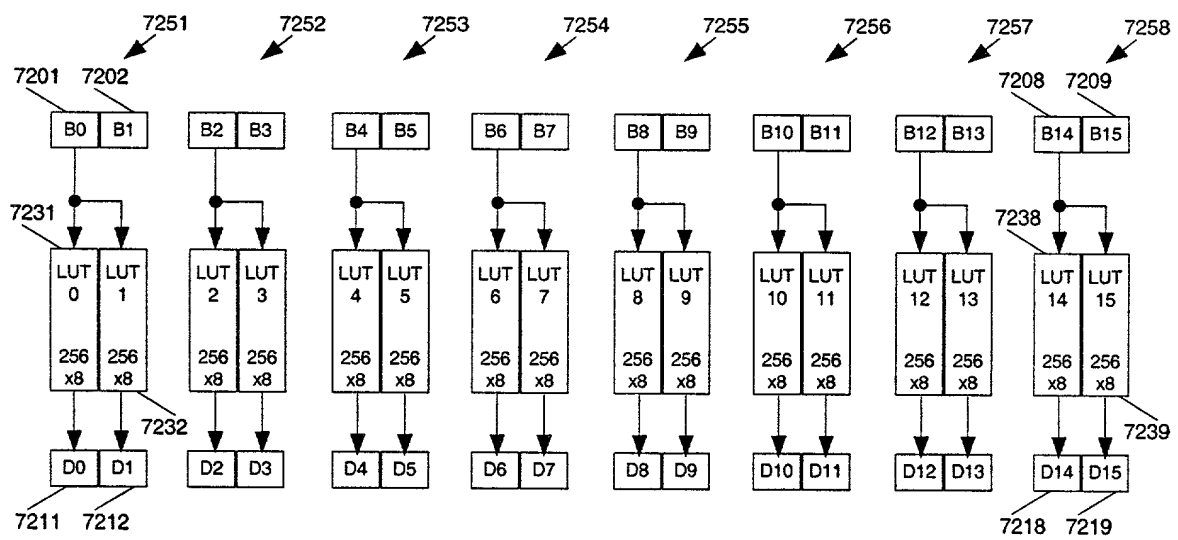


Fig. 52

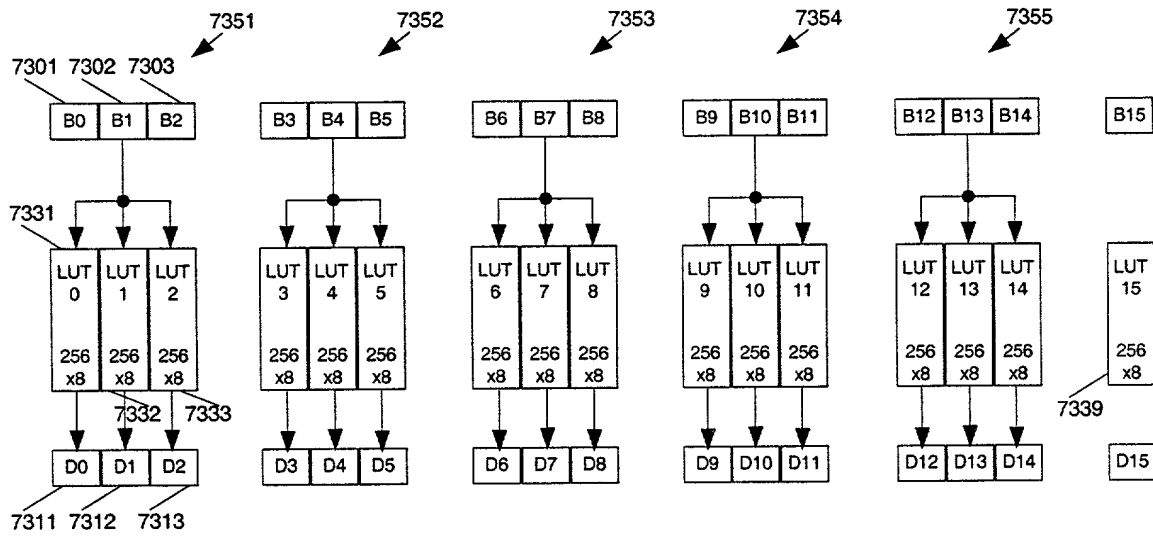


Fig. 53

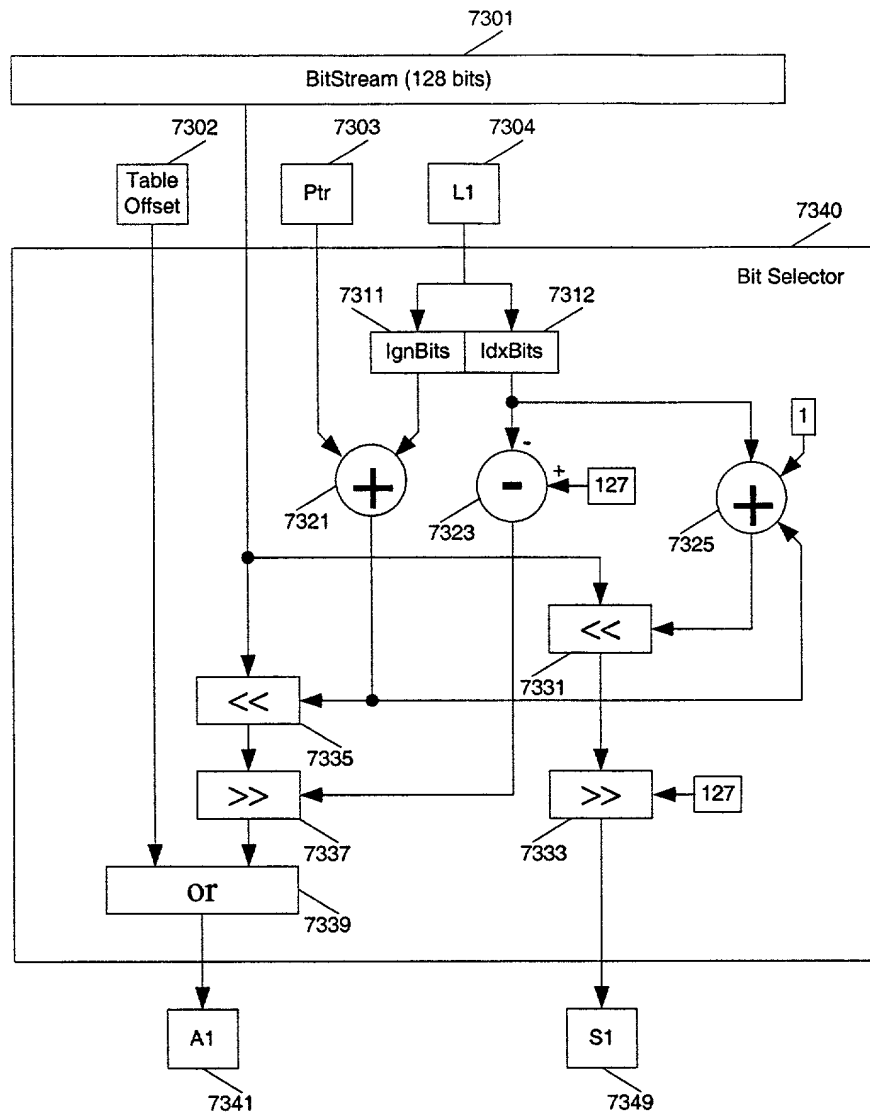


Fig. 54

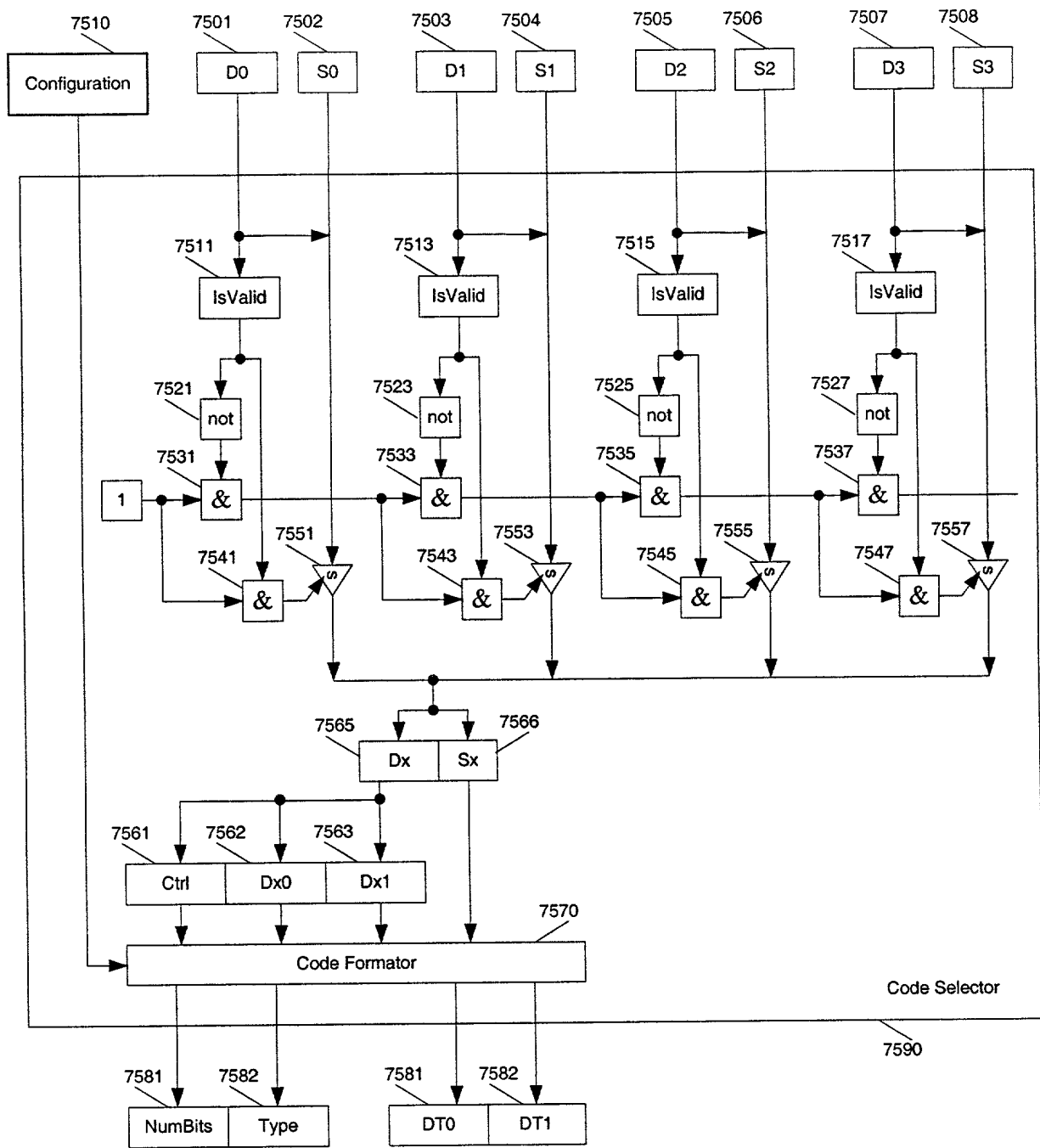


Fig. 55

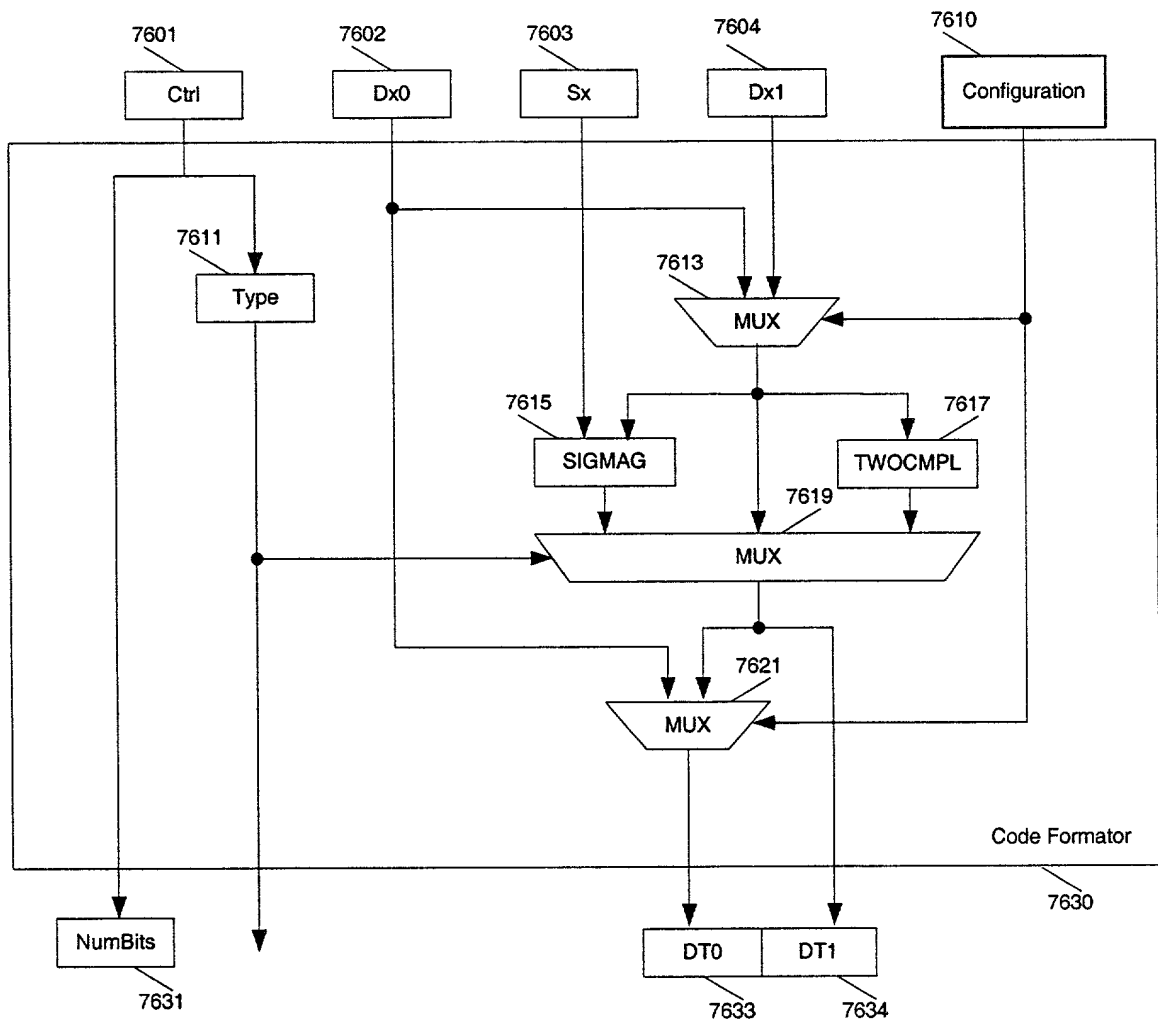


Fig. 56

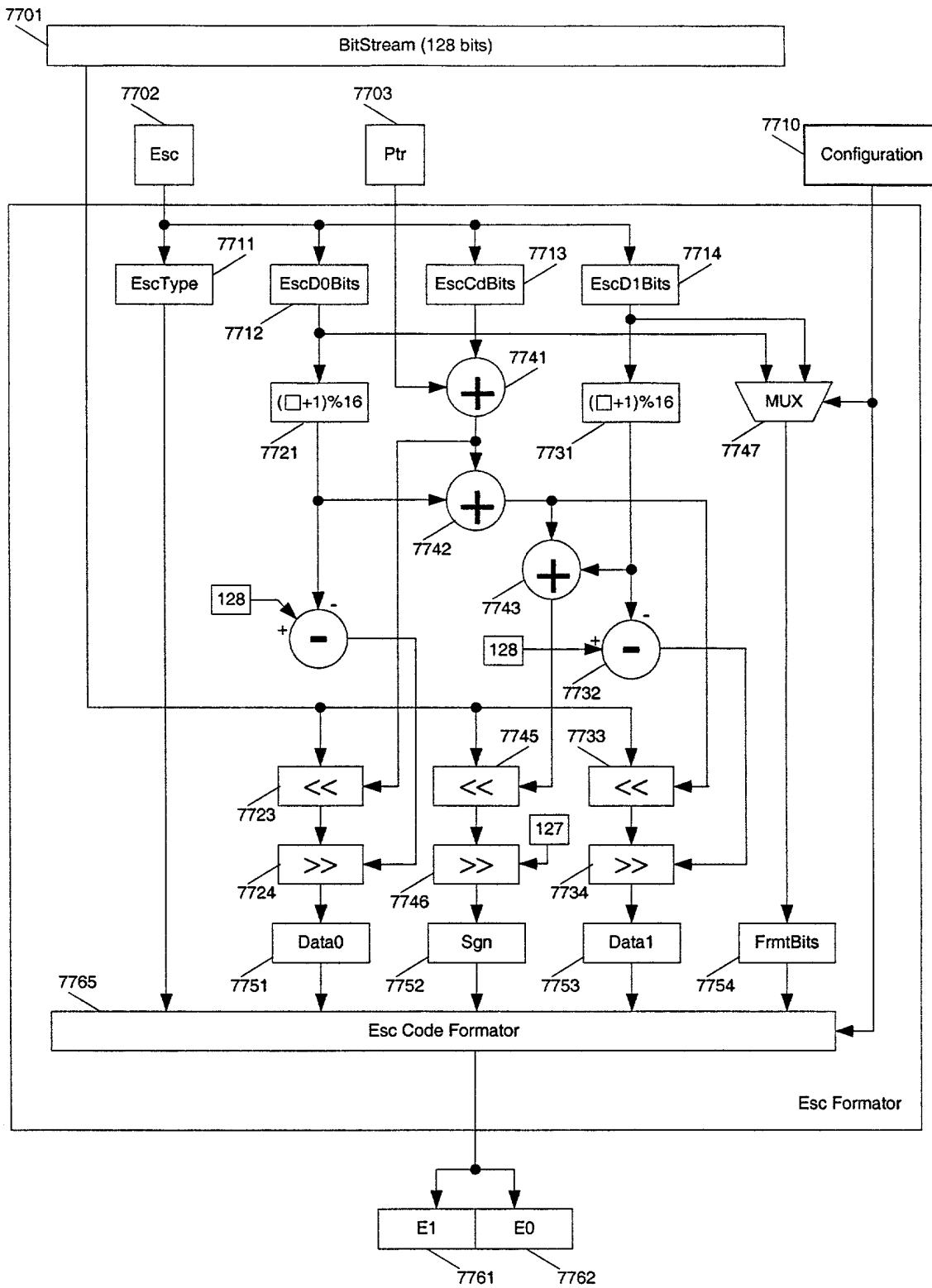


Fig. 57

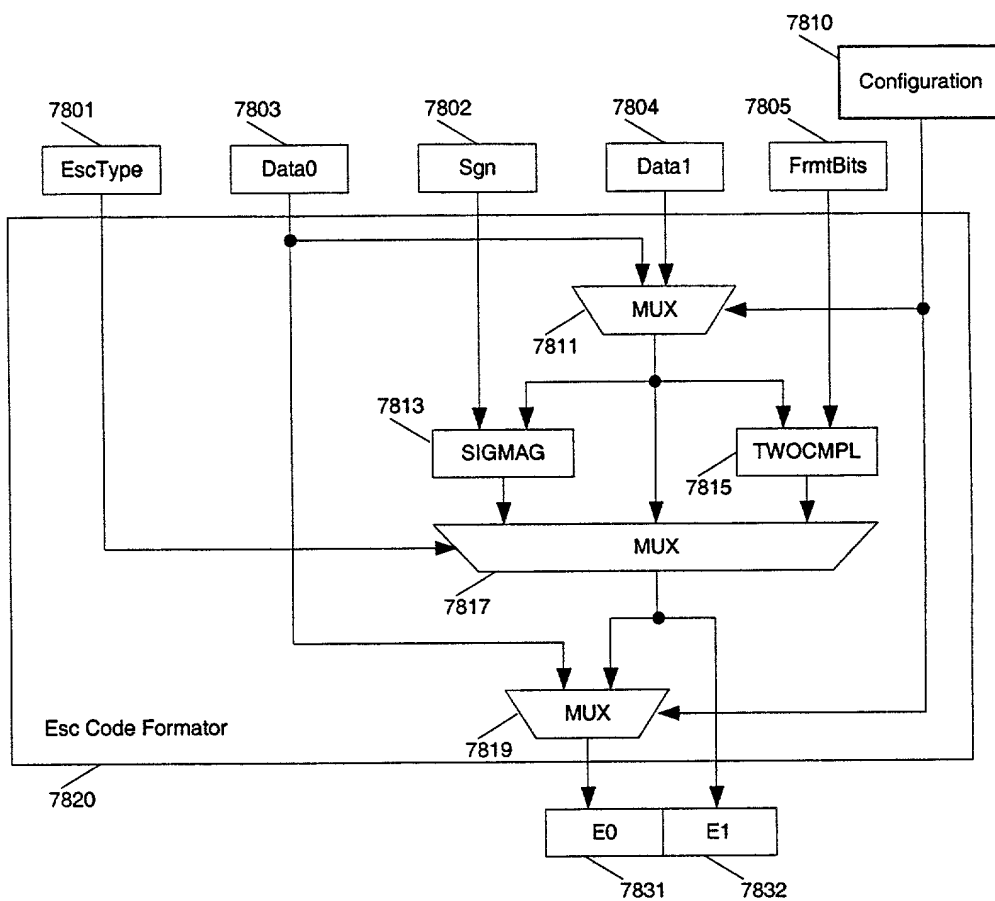


Fig. 58

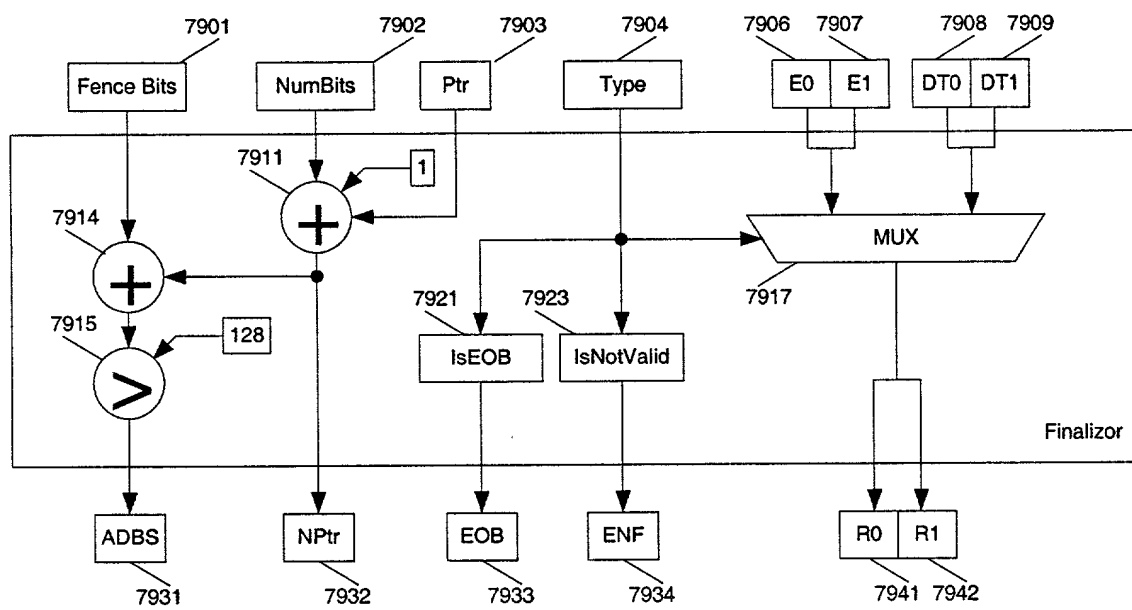


Fig. 59

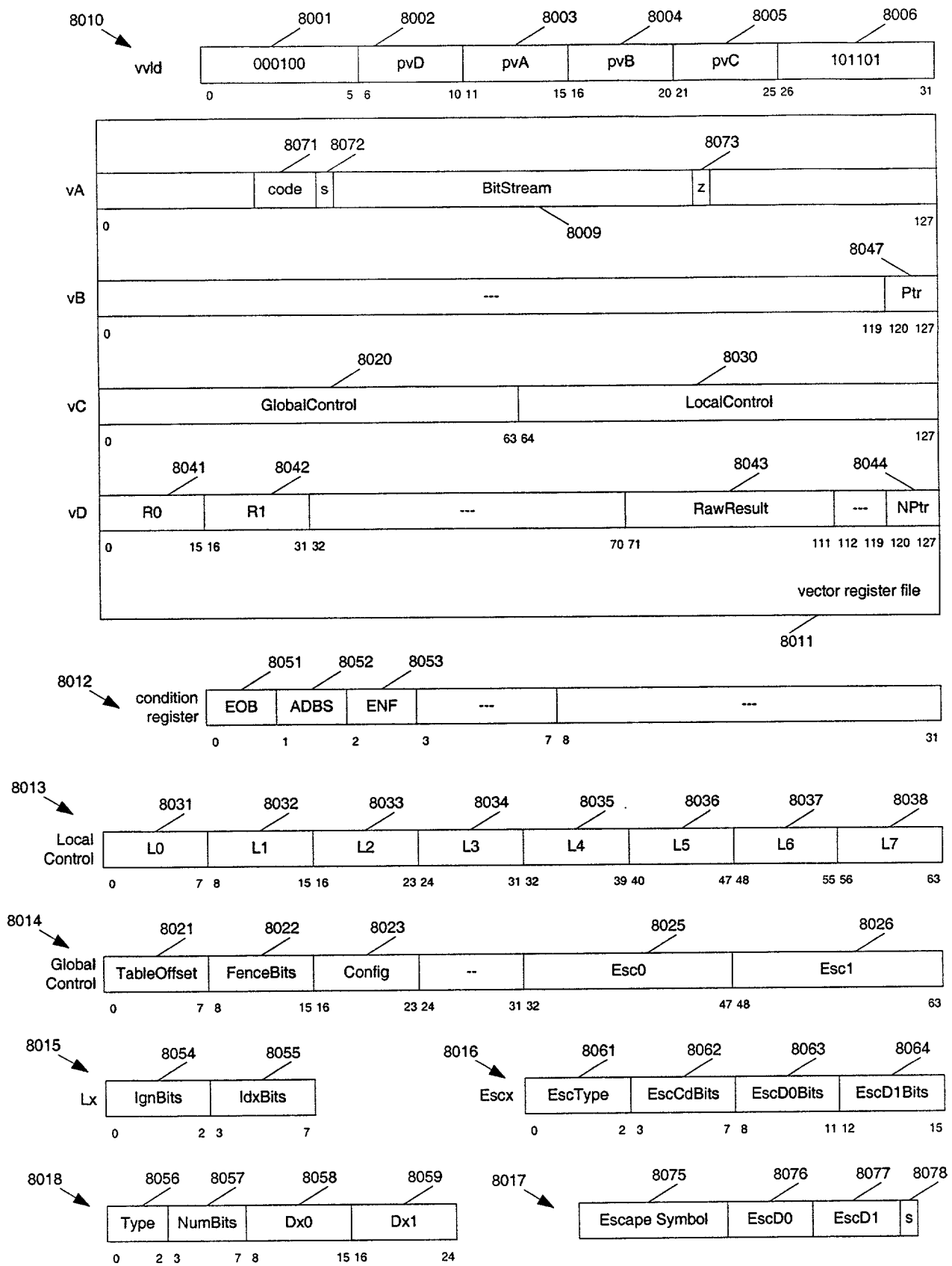


Fig. 60

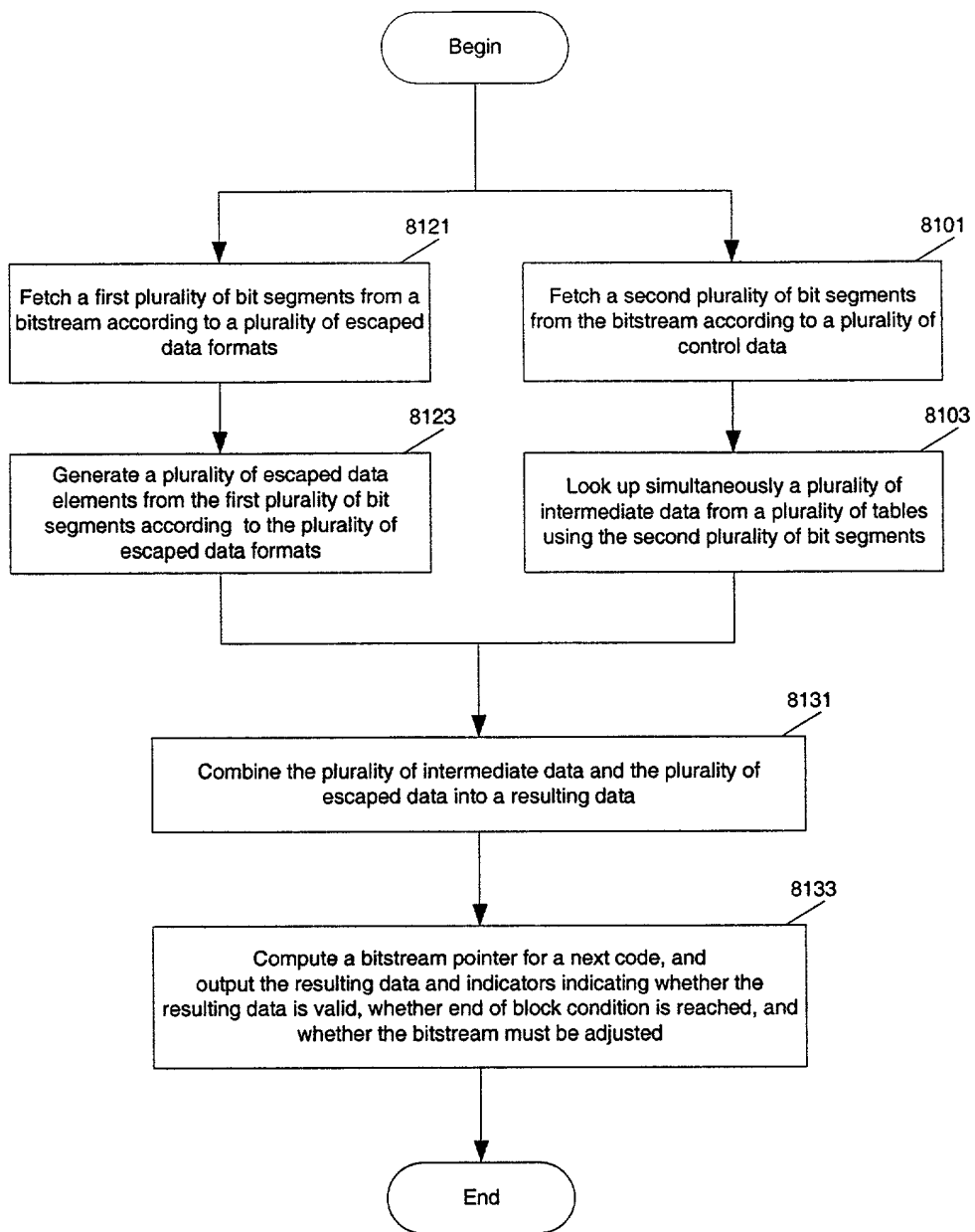


Fig. 61

| Index | T1 | | | | T2 | | | | T3 | | | | T4 | | | |
|-------|------|------|-----|-------|------|------|-----|-------|------|------|-----|-------|------|------|-----|-------|
| | type | bits | run | level | type | bits | run | level | type | bits | run | level | type | bits | run | level |
| 00 | 4 | 2 | 0 | 1 | 4 | 6 | 3 | 1 | 4 | 9 | 11 | 1 | 4 | 10 | 5 | 3 |
| 01 | 4 | 2 | 0 | 1 | 4 | 6 | 3 | 1 | 4 | 9 | 11 | -1 | 4 | 10 | 5 | 3 |
| 02 | 4 | 2 | 0 | 1 | 4 | 6 | 3 | -1 | 4 | 9 | 12 | 1 | 4 | 10 | 5 | -3 |
| 03 | 4 | 2 | 0 | 1 | 4 | 6 | 3 | -1 | 4 | 9 | 12 | -1 | 4 | 10 | 5 | -3 |
| 04 | 4 | 2 | 0 | 1 | 4 | 6 | 4 | 1 | 4 | 9 | 13 | 1 | 4 | 10 | 3 | 4 |
| 05 | 4 | 2 | 0 | 1 | 4 | 6 | 4 | 1 | 4 | 9 | 13 | -1 | 4 | 10 | 3 | 4 |
| 06 | 4 | 2 | 0 | 1 | 4 | 6 | 4 | -1 | 4 | 9 | 14 | 1 | 4 | 10 | 3 | -4 |
| 07 | 4 | 2 | 0 | 1 | 4 | 6 | 4 | -1 | 4 | 9 | 14 | -1 | 4 | 10 | 3 | -4 |
| 08 | 4 | 2 | 0 | -1 | 4 | 6 | 0 | 7 | 4 | 9 | 5 | 2 | 4 | 10 | 3 | 5 |
| 09 | 4 | 2 | 0 | -1 | 4 | 6 | 0 | 7 | 4 | 9 | 5 | -2 | 4 | 10 | 3 | 5 |
| 0a | 4 | 2 | 0 | -1 | 4 | 6 | 0 | -7 | 4 | 9 | 6 | 2 | 4 | 10 | 3 | -5 |
| 0b | 4 | 2 | 0 | -1 | 4 | 6 | 0 | -7 | 4 | 9 | 6 | -2 | 4 | 10 | 3 | -5 |
| 0c | 4 | 2 | 0 | -1 | 4 | 6 | 0 | 8 | 4 | 9 | 3 | 3 | 4 | 10 | 2 | 6 |
| 0d | 4 | 2 | 0 | -1 | 4 | 6 | 0 | 8 | 4 | 9 | 3 | -3 | 4 | 10 | 2 | 6 |
| 0e | 4 | 2 | 0 | -1 | 4 | 6 | 0 | -8 | 4 | 9 | 4 | 3 | 4 | 10 | 2 | -6 |
| 0f | 4 | 2 | 0 | -1 | 4 | 6 | 0 | -8 | 4 | 9 | 4 | -3 | 4 | 10 | 2 | -6 |
| 10 | 4 | 3 | 0 | 2 | 4 | 7 | 5 | 1 | 4 | 9 | 2 | 4 | 4 | 10 | 1 | 9 |
| 11 | 4 | 3 | 0 | 2 | 4 | 7 | 5 | -1 | 4 | 9 | 2 | -4 | 4 | 10 | 1 | 9 |
| 12 | 4 | 3 | 0 | 2 | 4 | 7 | 6 | 1 | 4 | 9 | 2 | 5 | 4 | 10 | 1 | -9 |
| 13 | 4 | 3 | 0 | 2 | 4 | 7 | 6 | -1 | 4 | 9 | 2 | -5 | 4 | 10 | 1 | -9 |
| 14 | 4 | 3 | 0 | -2 | 4 | 7 | 2 | 2 | 4 | 9 | 1 | 8 | 4 | 10 | 1 | 10 |
| 15 | 4 | 3 | 0 | -2 | 4 | 7 | 2 | -2 | 4 | 9 | 1 | -8 | 4 | 10 | 1 | 10 |
| 16 | 4 | 3 | 0 | -2 | 4 | 7 | 1 | 3 | 4 | 9 | 0 | 18 | 4 | 10 | 1 | -10 |
| 17 | 4 | 3 | 0 | -2 | 4 | 7 | 1 | -3 | 4 | 9 | 0 | -18 | 4 | 10 | 1 | -10 |
| 18 | 1 | 3 | 0 | 0 | 4 | 7 | 1 | 4 | 4 | 9 | 0 | 19 | 4 | 10 | 1 | 11 |
| 19 | 1 | 3 | 0 | 0 | 4 | 7 | 1 | -4 | 4 | 9 | 0 | -19 | 4 | 10 | 1 | 11 |
| 1a | 1 | 3 | 0 | 0 | 4 | 7 | 0 | 9 | 4 | 9 | 0 | 20 | 4 | 10 | 1 | -11 |
| 1b | 1 | 3 | 0 | 0 | 4 | 7 | 0 | -9 | 4 | 9 | 0 | -20 | 4 | 10 | 1 | -11 |
| 1c | 4 | 4 | 1 | 1 | 4 | 7 | 0 | 10 | 4 | 9 | 0 | 21 | 4 | 10 | 0 | 0 |
| 1d | 4 | 4 | 1 | 1 | 4 | 7 | 0 | -10 | 4 | 9 | 0 | -21 | 4 | 10 | 0 | 0 |
| 1e | 4 | 4 | 1 | -1 | 4 | 7 | 0 | 11 | 4 | 9 | 0 | 22 | 4 | 10 | 1 | 0 |
| 1f | 4 | 4 | 1 | -1 | 4 | 7 | 0 | -11 | 4 | 9 | 0 | -22 | 4 | 10 | 1 | 0 |
| 20 | 4 | 4 | 0 | 3 | 5 | 8 | 7 | 1 | 0 | 0 | 0 | 0 | 4 | 11 | 6 | 3 |
| 21 | 4 | 4 | 0 | 3 | 5 | 8 | 8 | 1 | 0 | 0 | 0 | 0 | 4 | 11 | 6 | -3 |
| 22 | 4 | 4 | 0 | -3 | 5 | 8 | 9 | 1 | 0 | 0 | 0 | 0 | 4 | 11 | 4 | 4 |
| 23 | 4 | 4 | 0 | -3 | 5 | 8 | 10 | 1 | 0 | 0 | 0 | 0 | 4 | 11 | 4 | -4 |
| 24 | 4 | 4 | 0 | 4 | 5 | 8 | 3 | 2 | 0 | 0 | 0 | 0 | 4 | 11 | 3 | 6 |
| 25 | 4 | 4 | 0 | 4 | 5 | 8 | 4 | 2 | 0 | 0 | 0 | 0 | 4 | 11 | 3 | -6 |
| 26 | 4 | 4 | 0 | -4 | 5 | 8 | 2 | 3 | 0 | 0 | 0 | 0 | 4 | 11 | 1 | 12 |
| 27 | 4 | 4 | 0 | -4 | 5 | 8 | 1 | 5 | 0 | 0 | 0 | 0 | 4 | 11 | 1 | -12 |
| 28 | 4 | 5 | 2 | 1 | 5 | 8 | 1 | 6 | 0 | 0 | 0 | 0 | 4 | 11 | 1 | 13 |
| 29 | 4 | 5 | 2 | -1 | 5 | 8 | 1 | 7 | 0 | 0 | 0 | 0 | 4 | 11 | 1 | -13 |
| 2a | 4 | 5 | 1 | 2 | 5 | 8 | 0 | 12 | 0 | 0 | 0 | 0 | 4 | 11 | 1 | 14 |
| 2b | 4 | 5 | 1 | -2 | 5 | 8 | 0 | 13 | 0 | 0 | 0 | 0 | 4 | 11 | 1 | -14 |
| 2c | 4 | 5 | 0 | 5 | 5 | 8 | 0 | 14 | 0 | 0 | 0 | 0 | 4 | 11 | 2 | 0 |
| 2d | 4 | 5 | 0 | -5 | 5 | 8 | 0 | 15 | 0 | 0 | 0 | 0 | 4 | 11 | 3 | 0 |
| 2e | 4 | 5 | 0 | 6 | 5 | 8 | 0 | 16 | 0 | 0 | 0 | 0 | 4 | 11 | 4 | 0 |
| 2f | 4 | 5 | 0 | -6 | 5 | 8 | 0 | 17 | 0 | 0 | 0 | 0 | 4 | 11 | 5 | 0 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 7 | 2 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 8 | 2 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 9 | 2 |
| 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 10 | 2 |
| 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 7 | 3 |
| 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 8 | 3 |
| 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 4 | 5 |
| 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 3 | 7 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 2 | 7 |
| 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 2 | 8 |
| 3a | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 2 | 9 |
| 3b | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 2 | 10 |
| 3c | 0 | 0 | 0 | 0 | 2 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 2 | 11 |
| 3d | 0 | 0 | 0 | 0 | 2 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 1 | 15 |
| 3e | 0 | 0 | 0 | 0 | 3 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 1 | 16 |
| 3f | 0 | 0 | 0 | 0 | 3 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 12 | 1 | 17 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ff | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Fig. 62

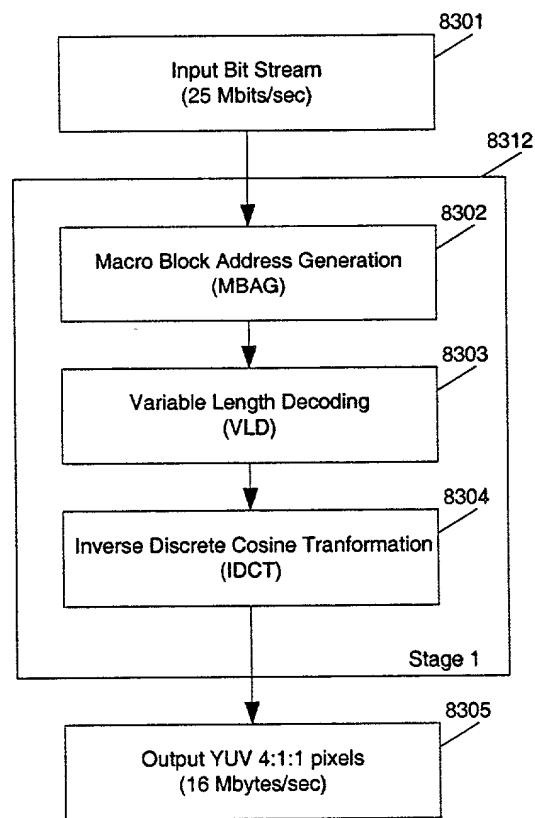


Fig. 63

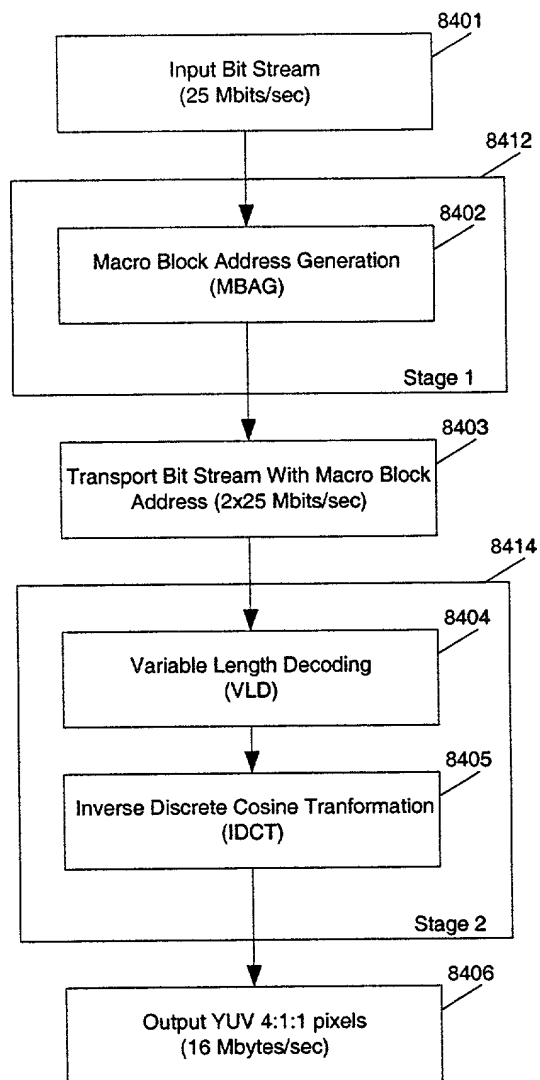


Fig. 64

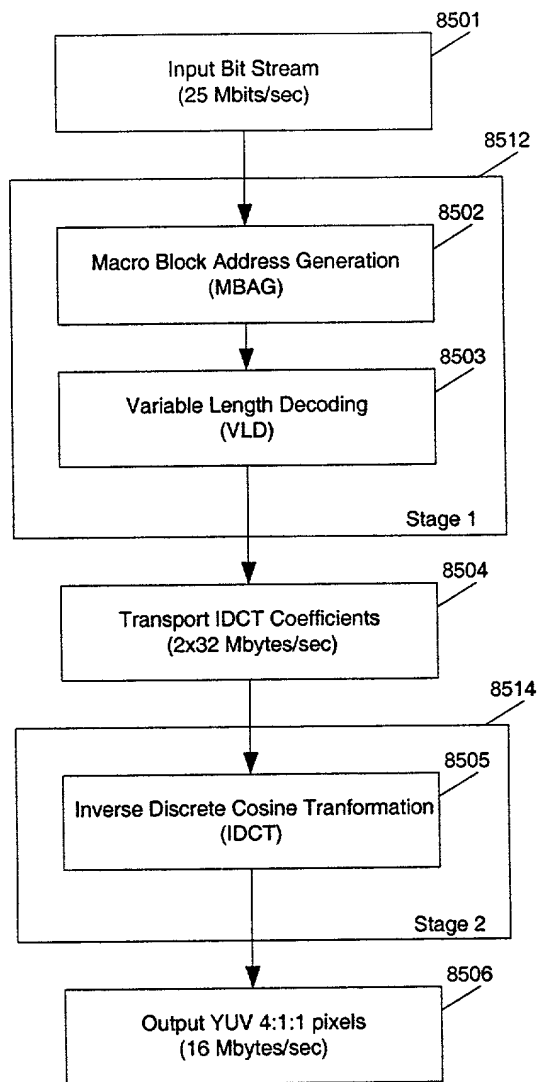


Fig. 65

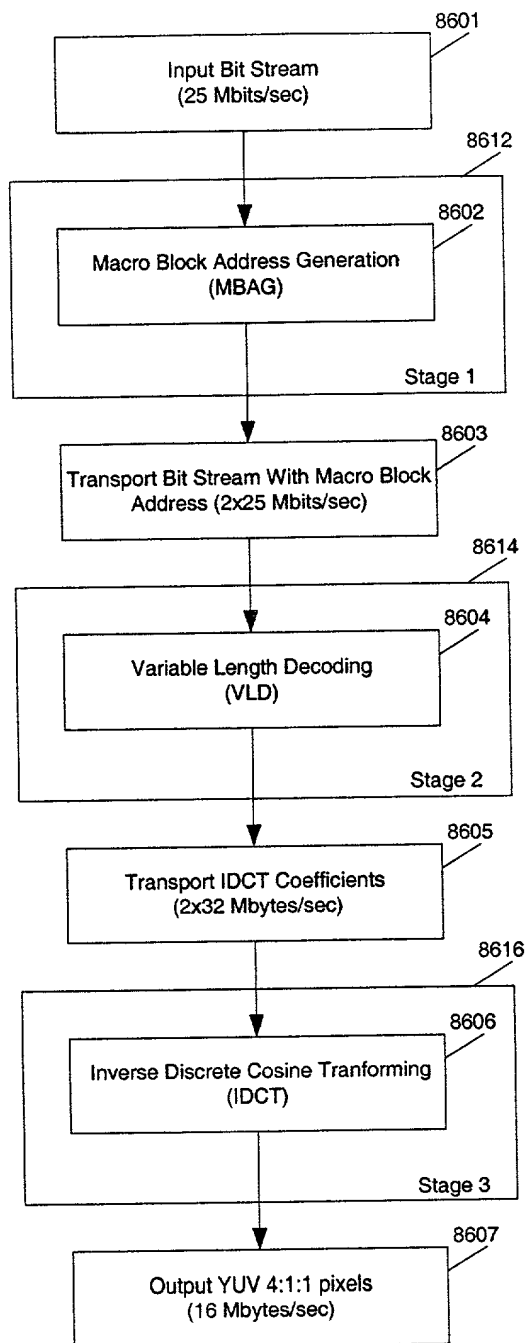


Fig. 66

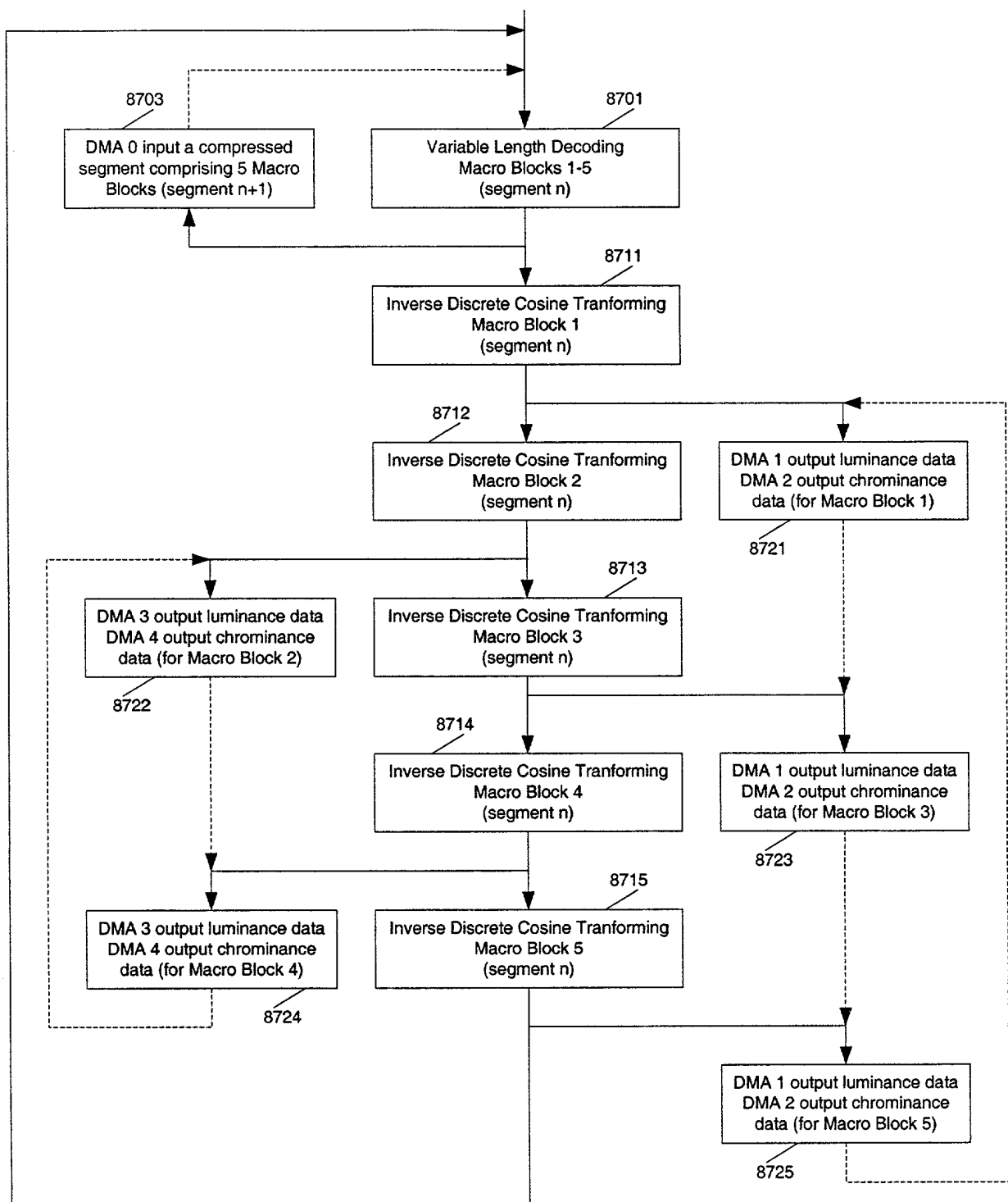


Fig. 67

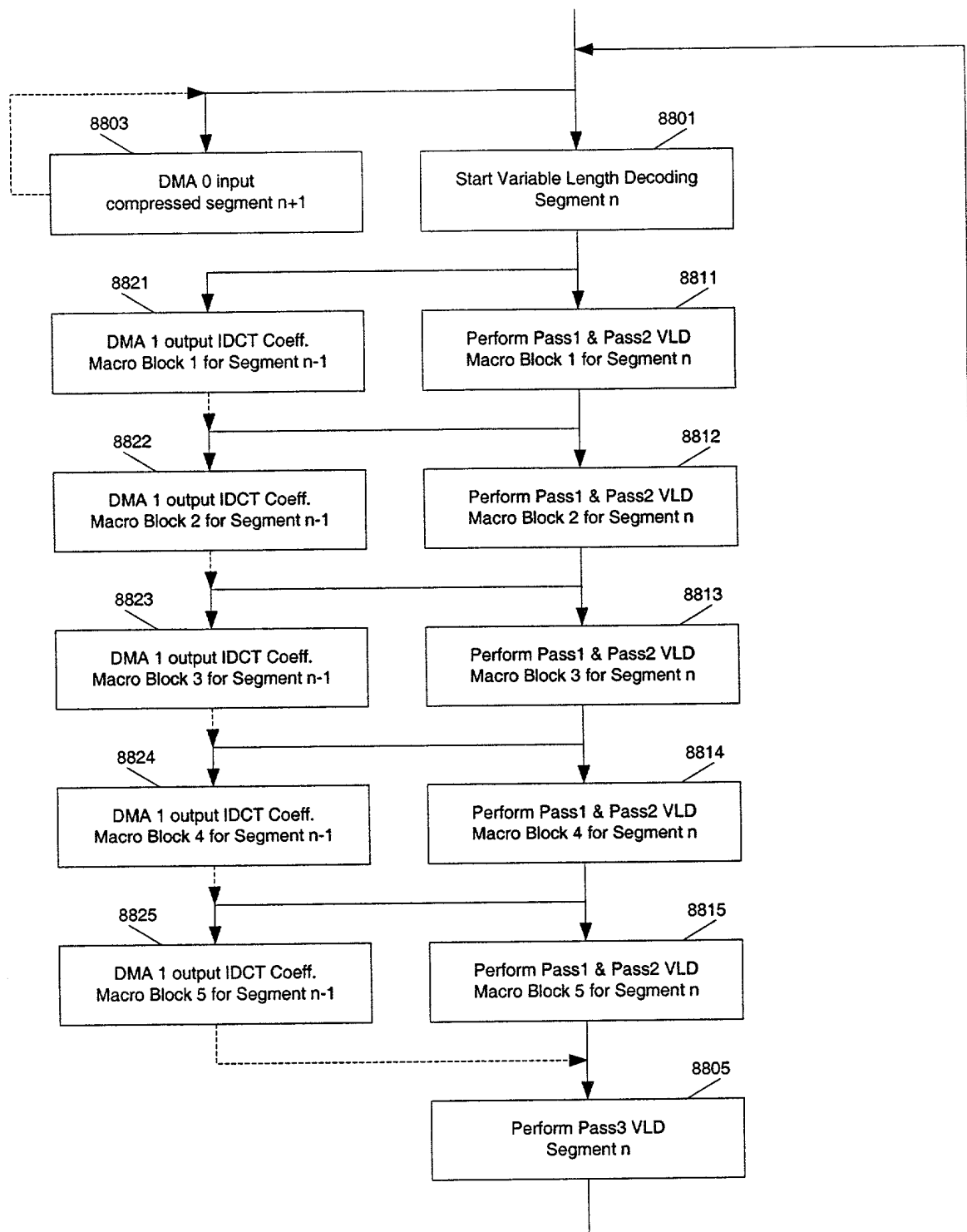


Fig. 68

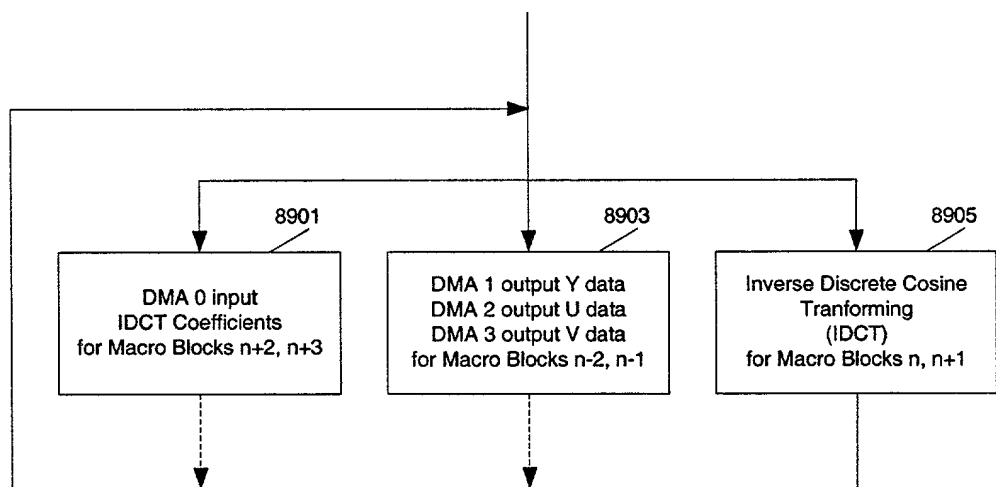


Fig. 69

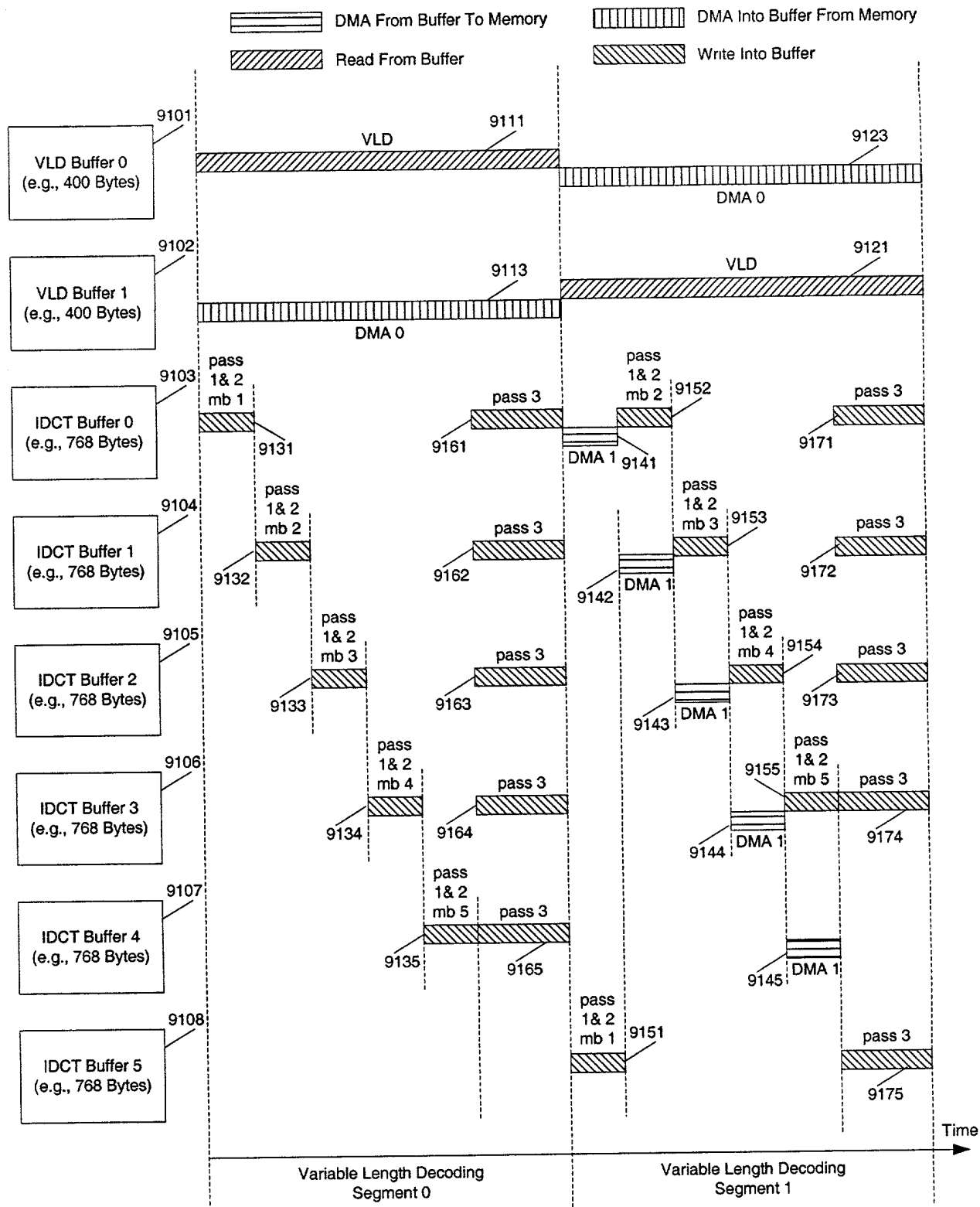


Fig. 71

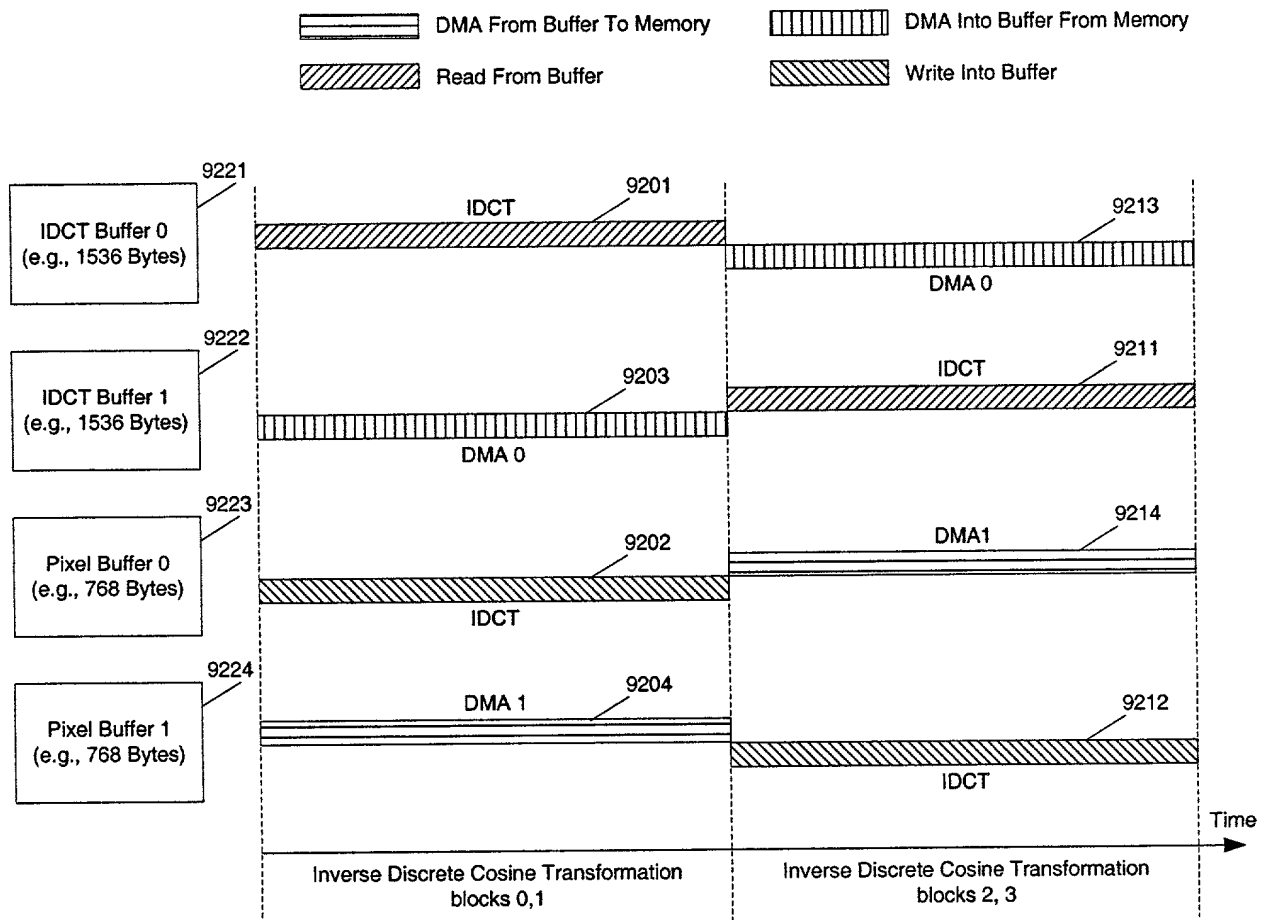


Fig. 72

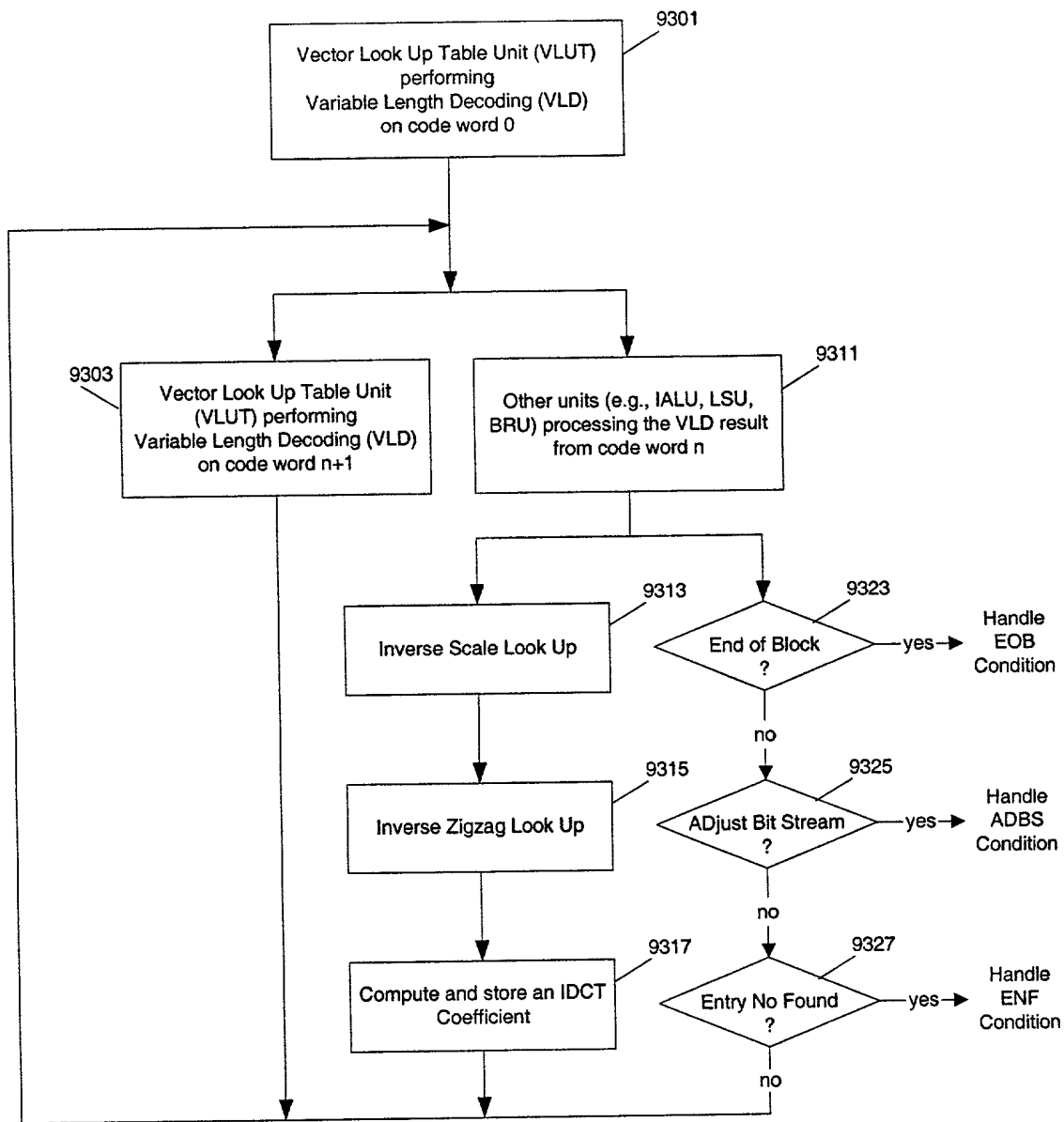


Fig. 73

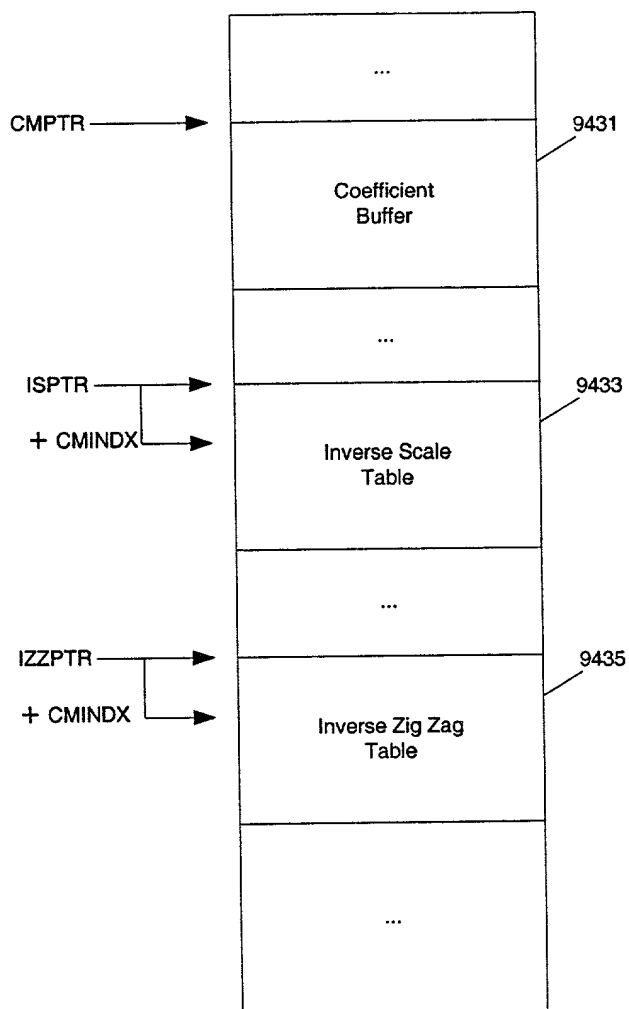
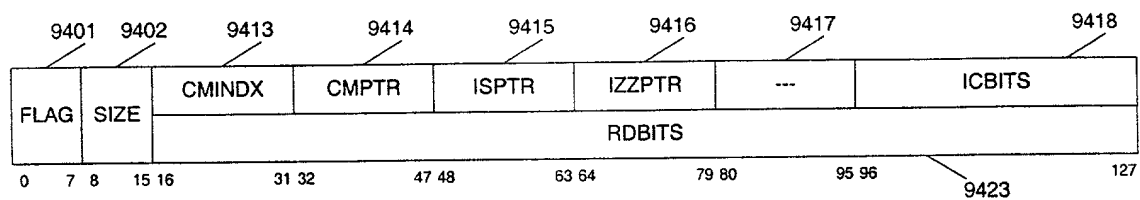


Fig. 74

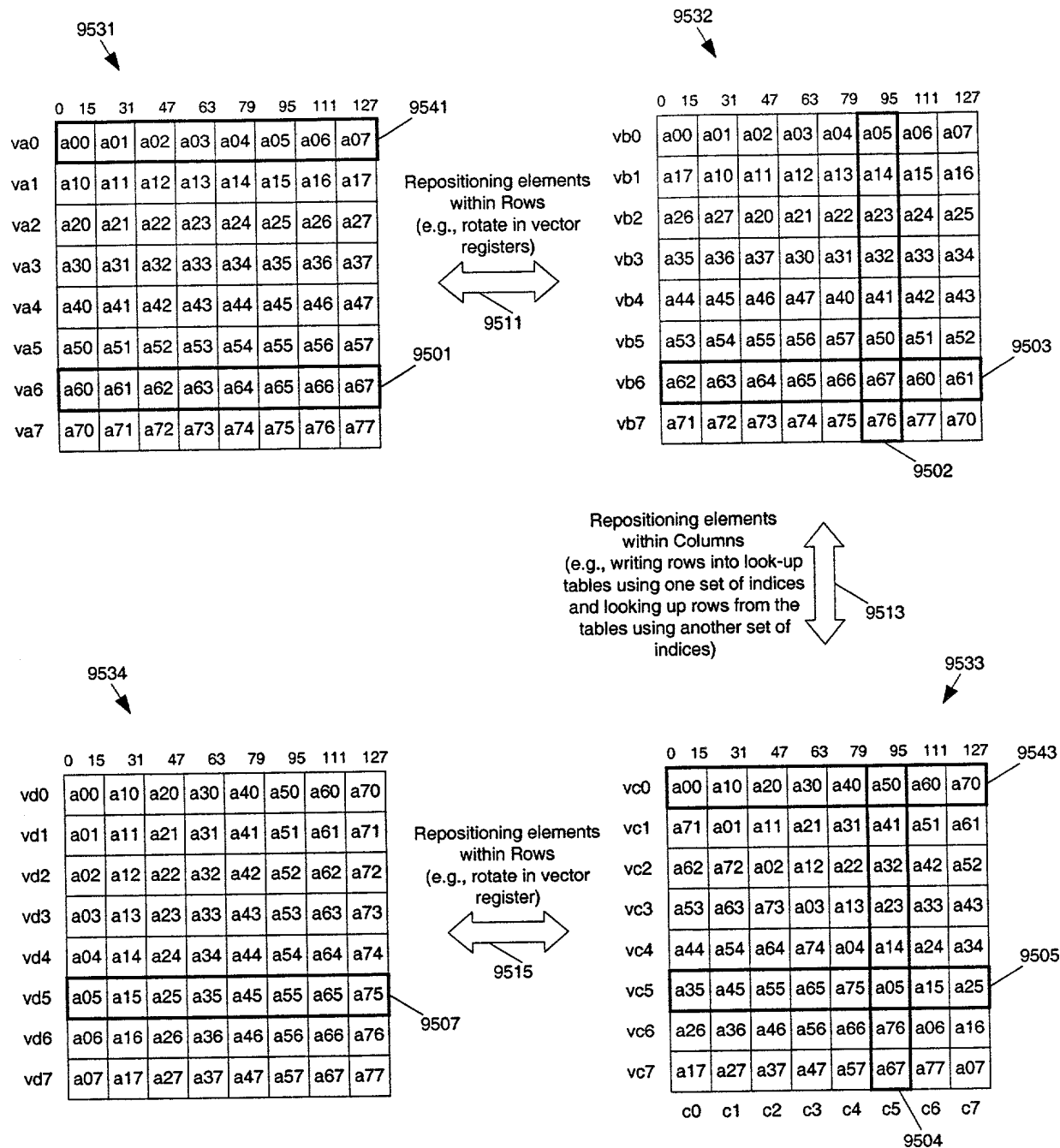


Fig. 75

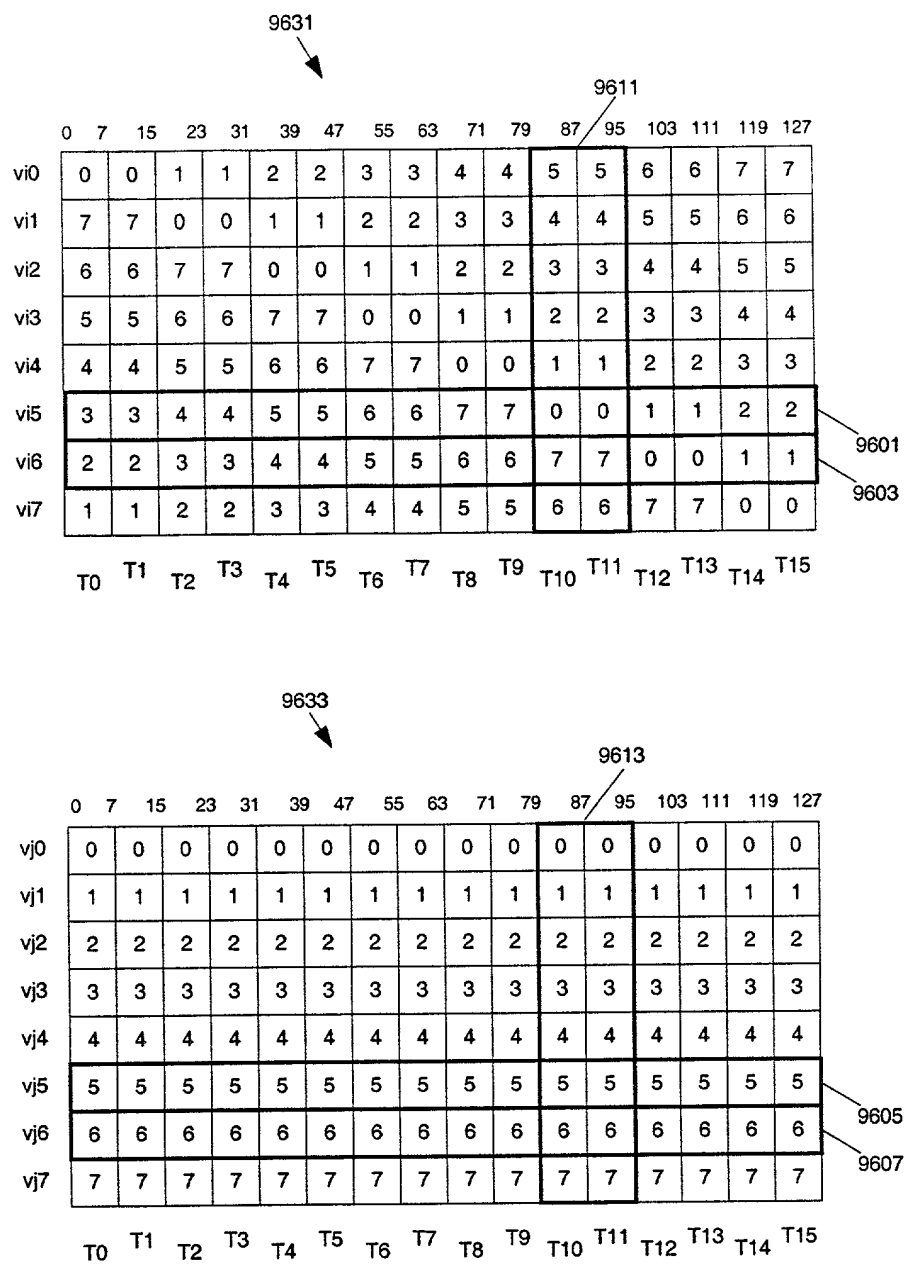


Fig. 76

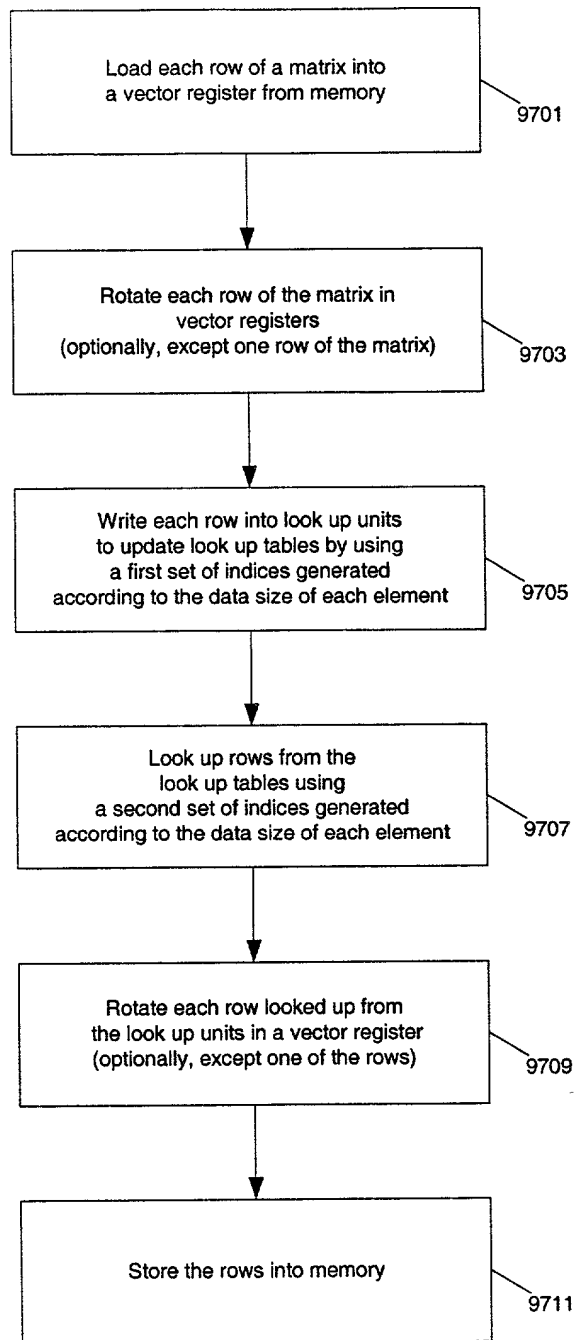


Fig. 77

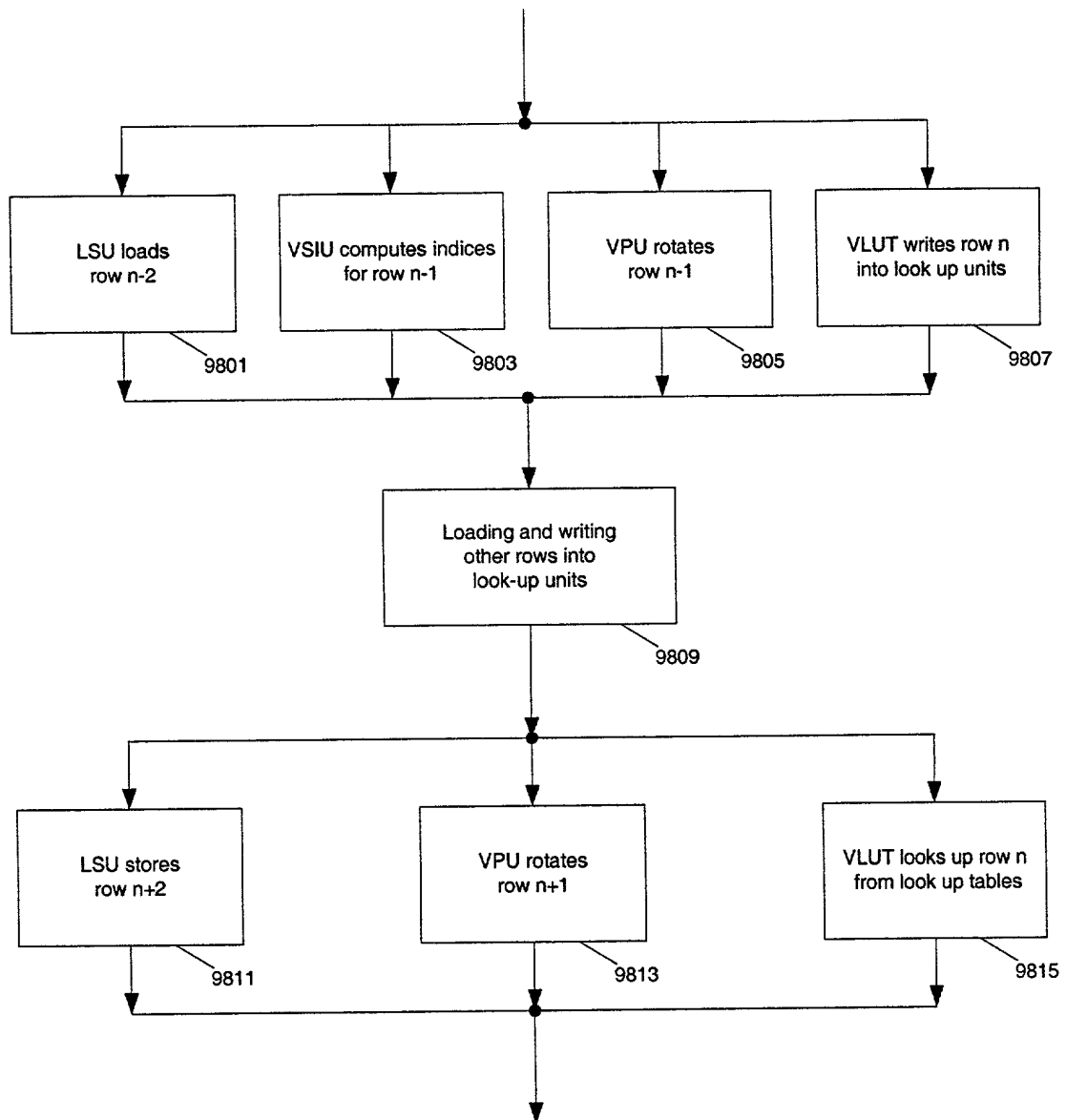


Fig. 78

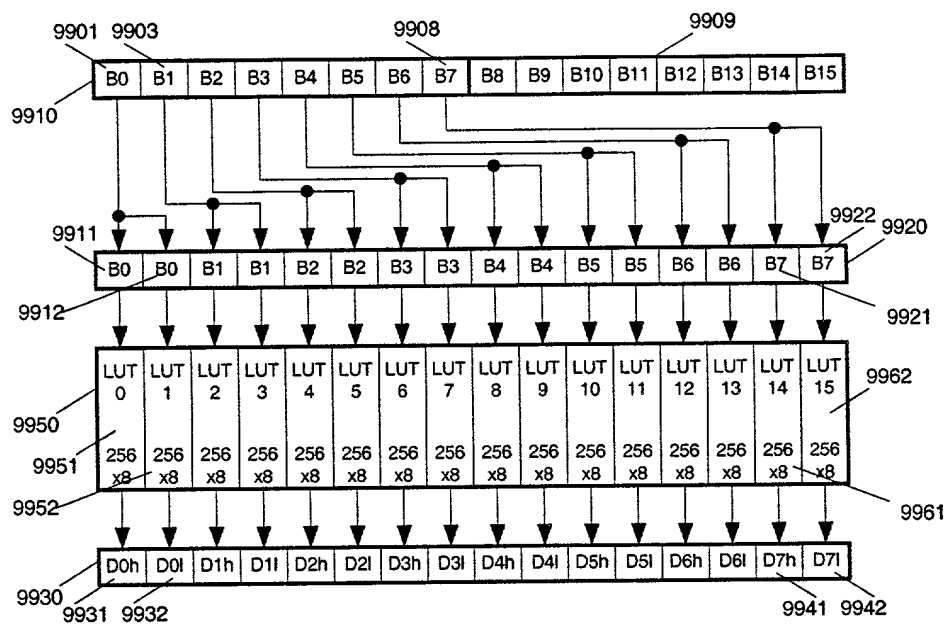


Fig. 79

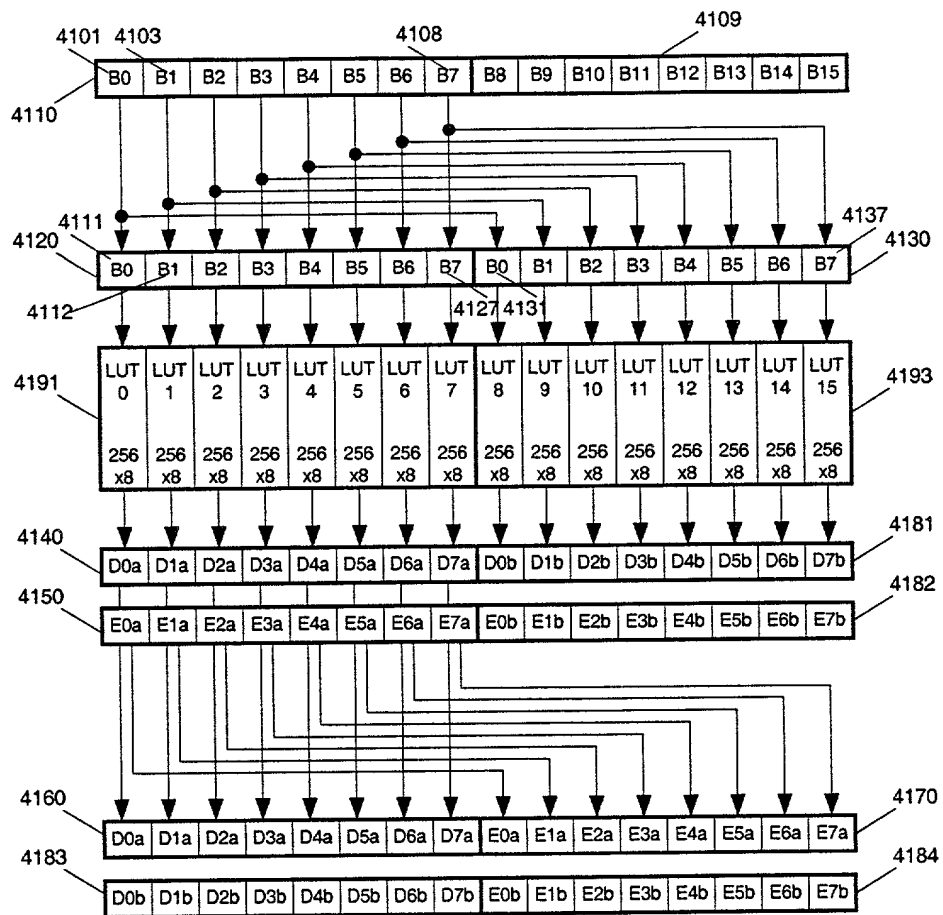


Fig. 80

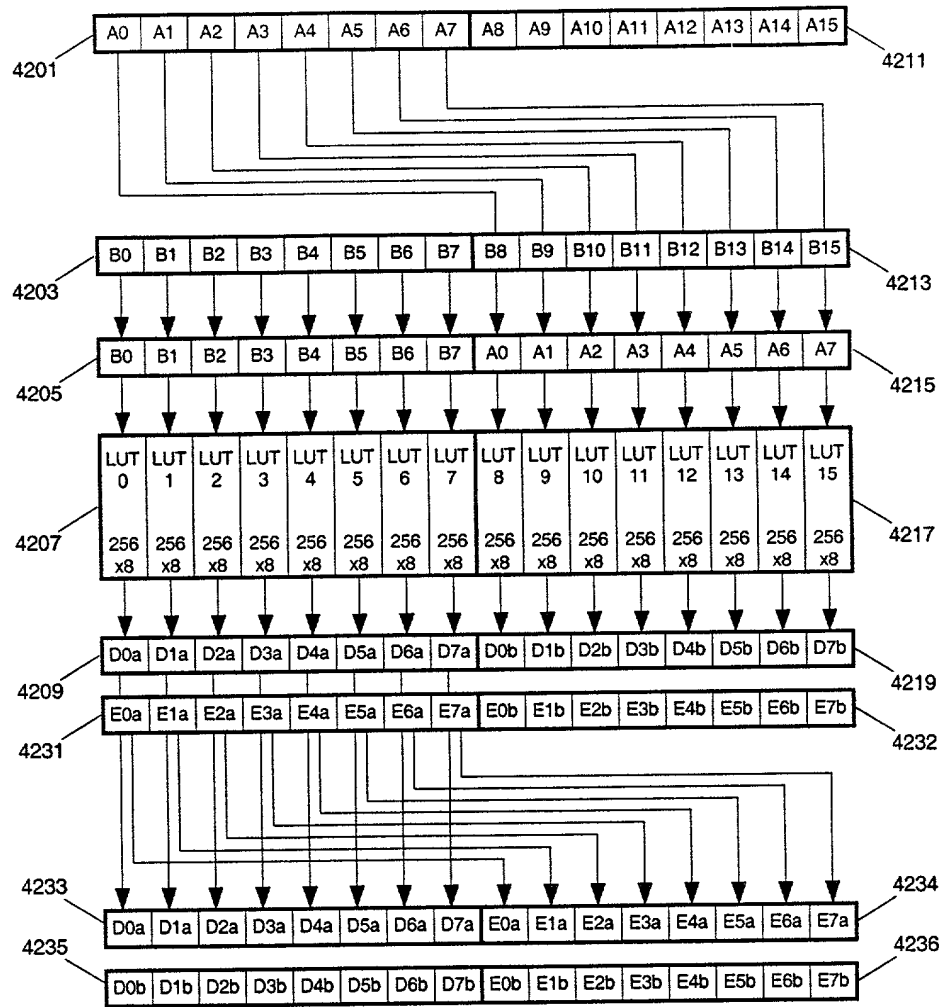


Fig. 81

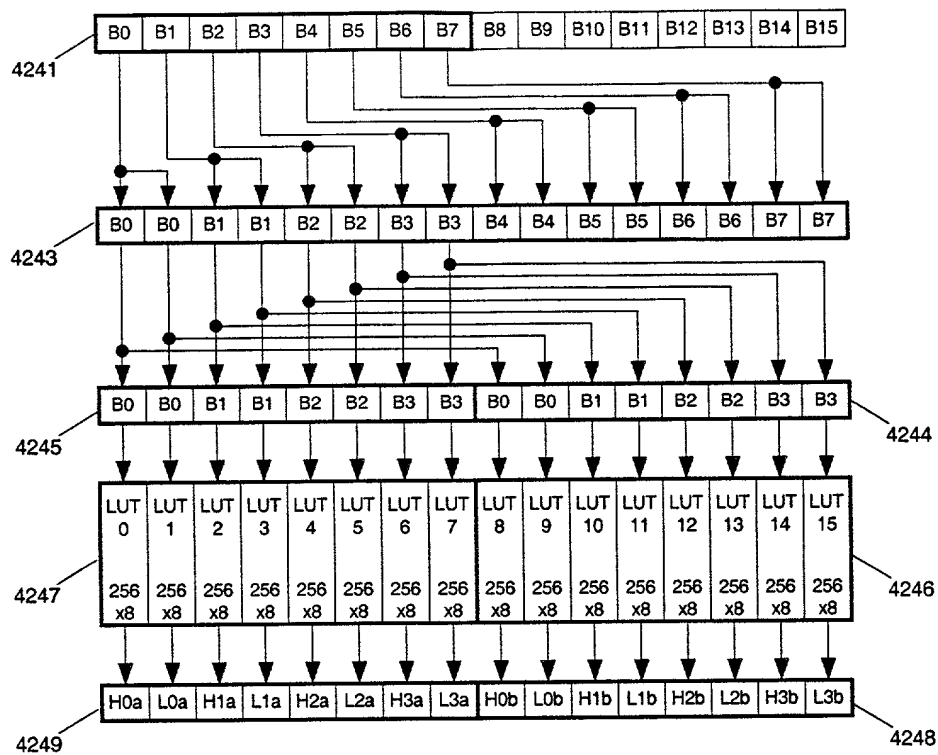


Fig. 82

Fig. 83

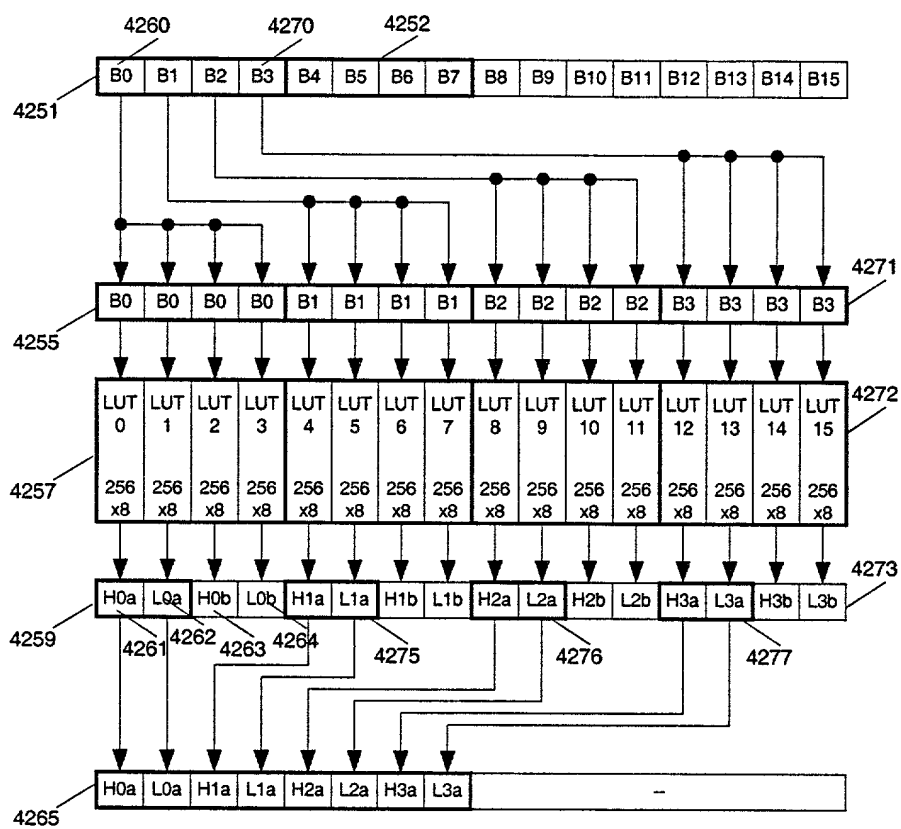


Fig. 83

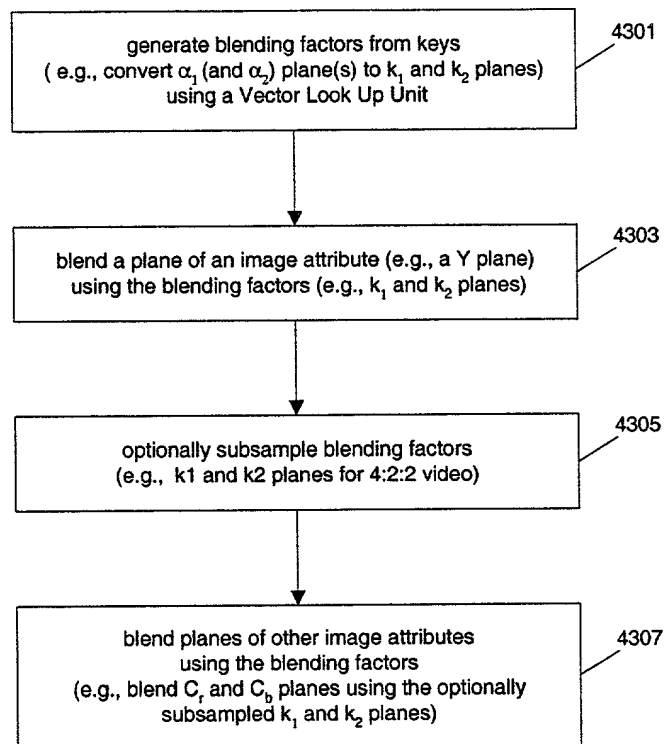


Fig. 84

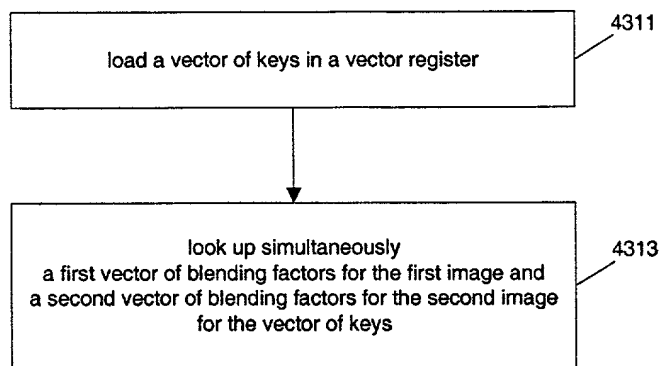


Fig. 85

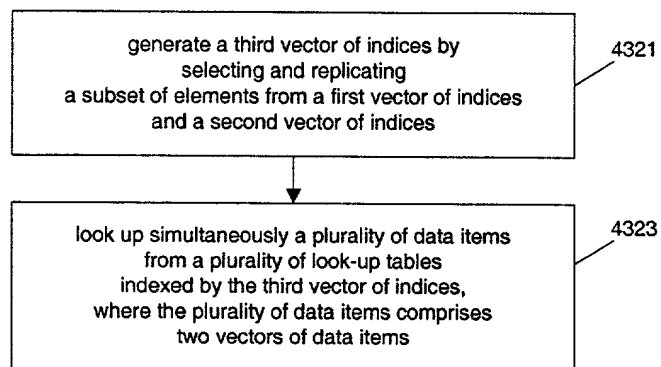


Fig. 86

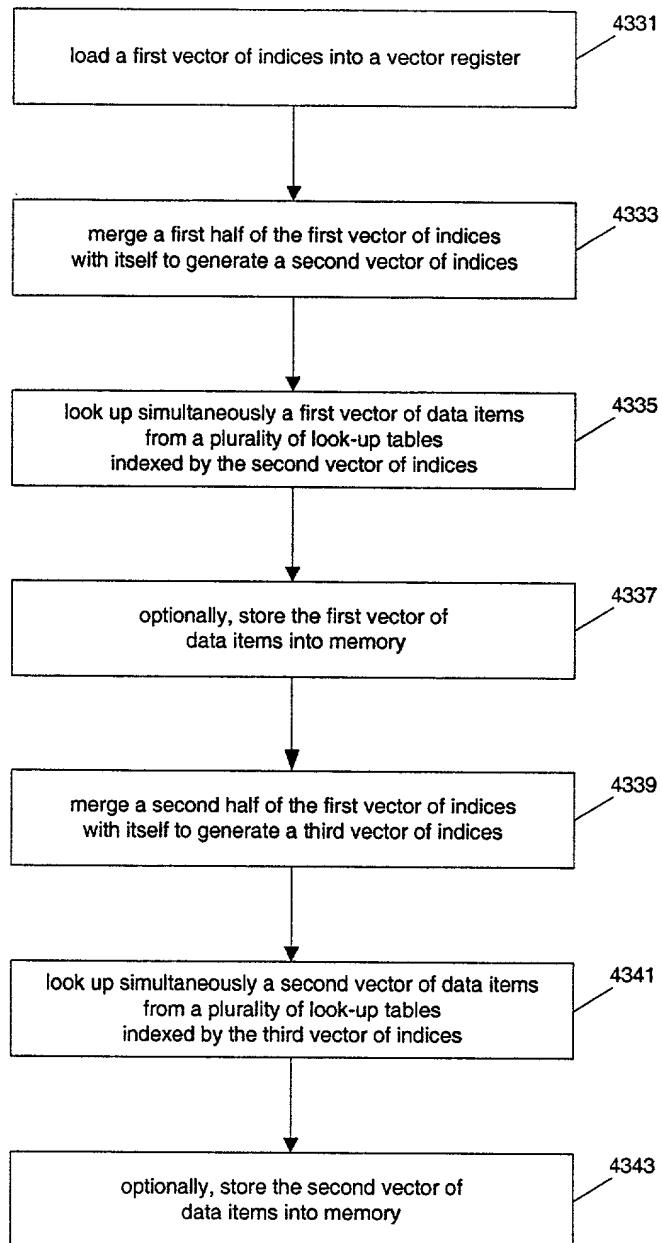


Fig. 87

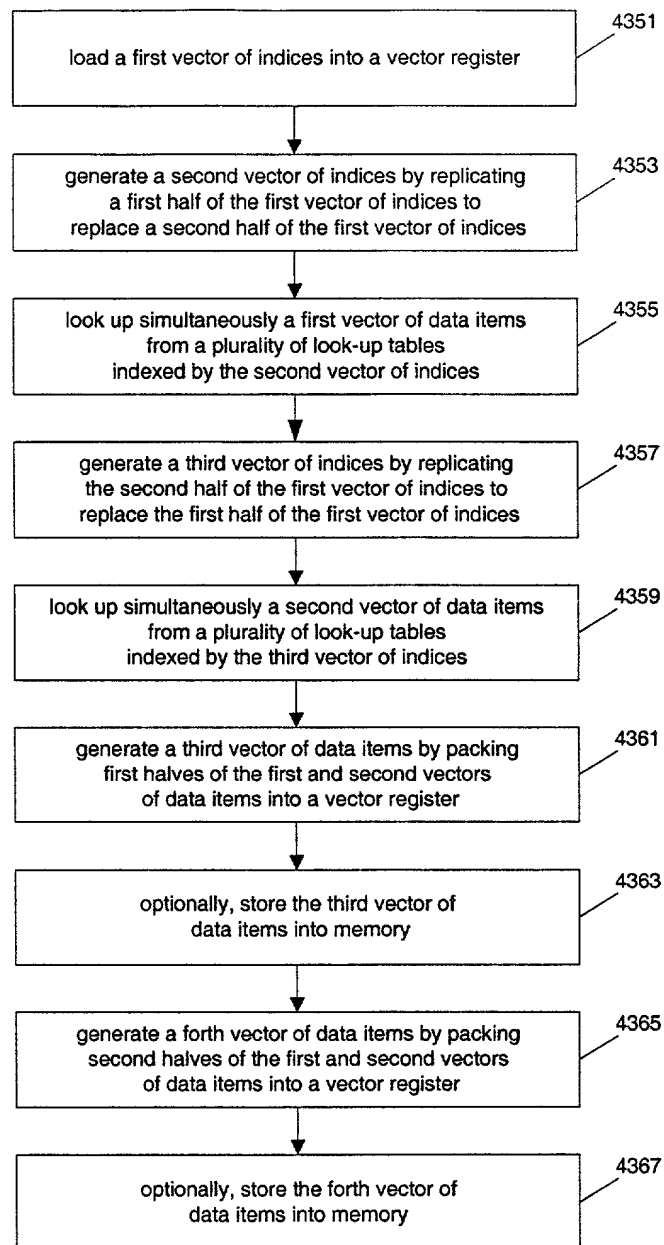


Fig. 88

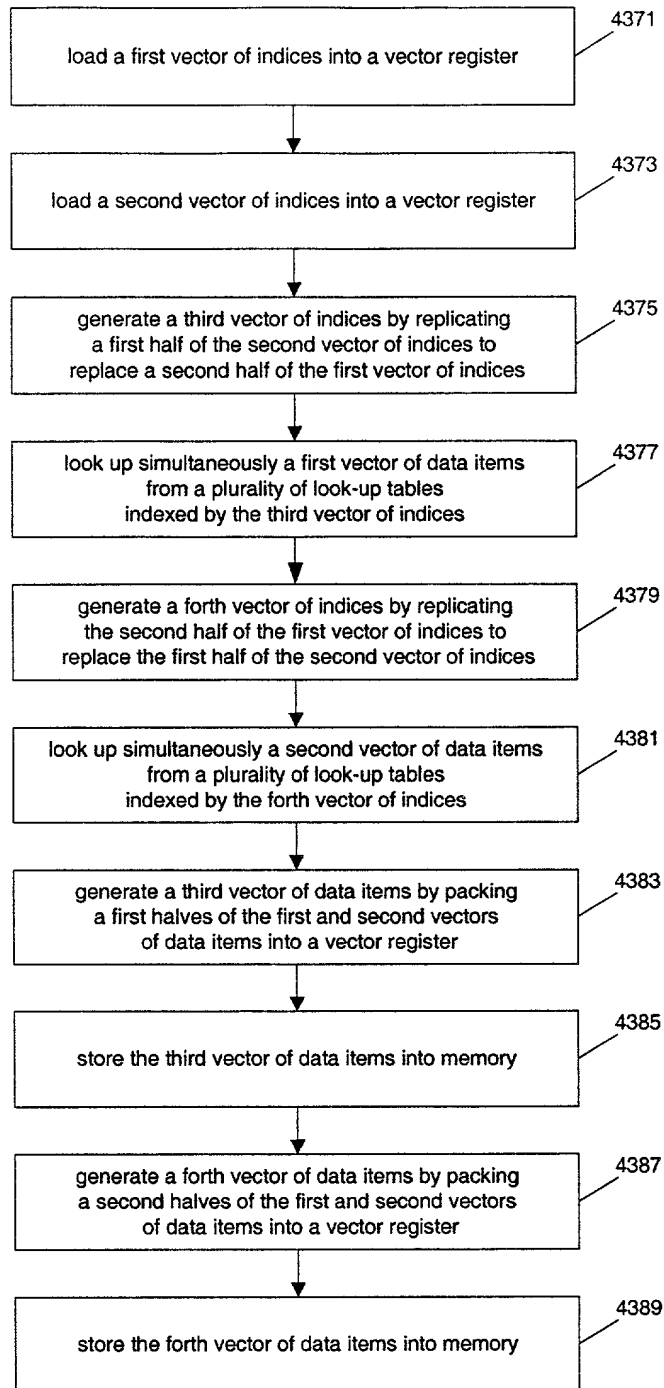


Fig. 89

Fig. 90

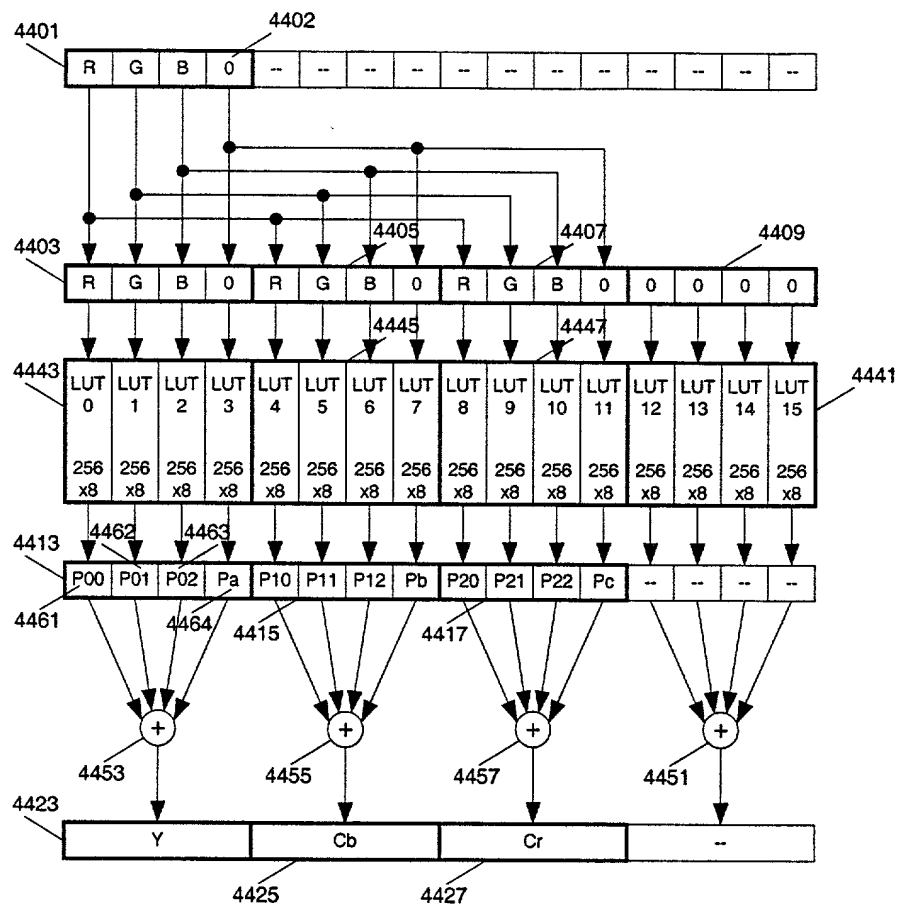


Fig. 90

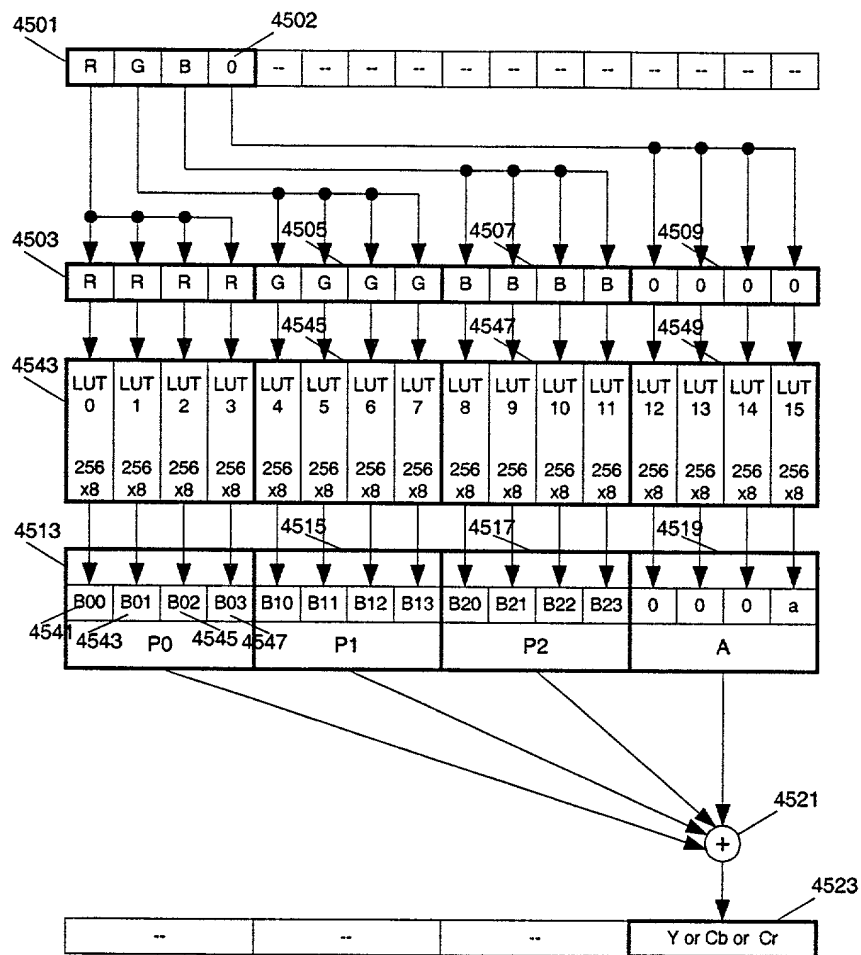


Fig. 91

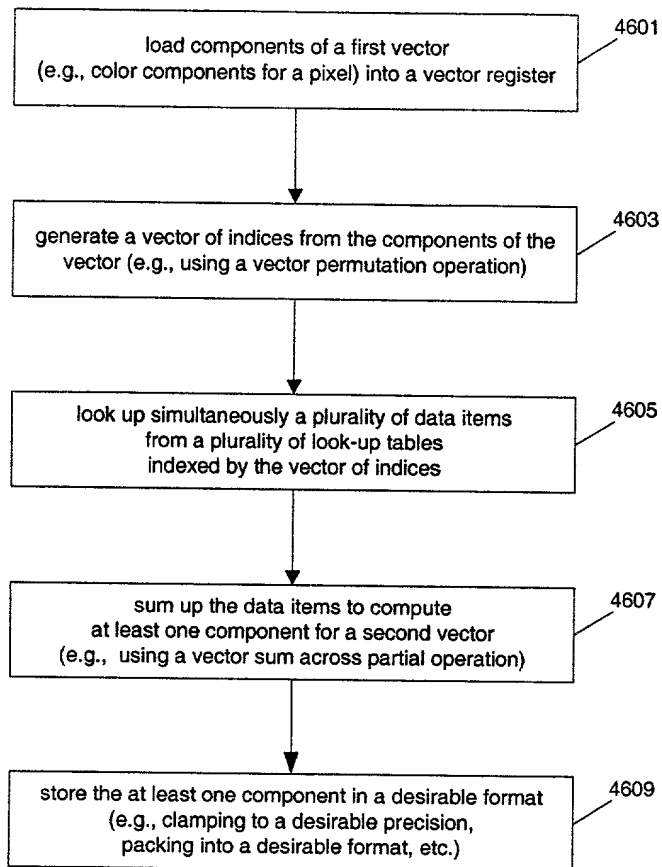


Fig. 92

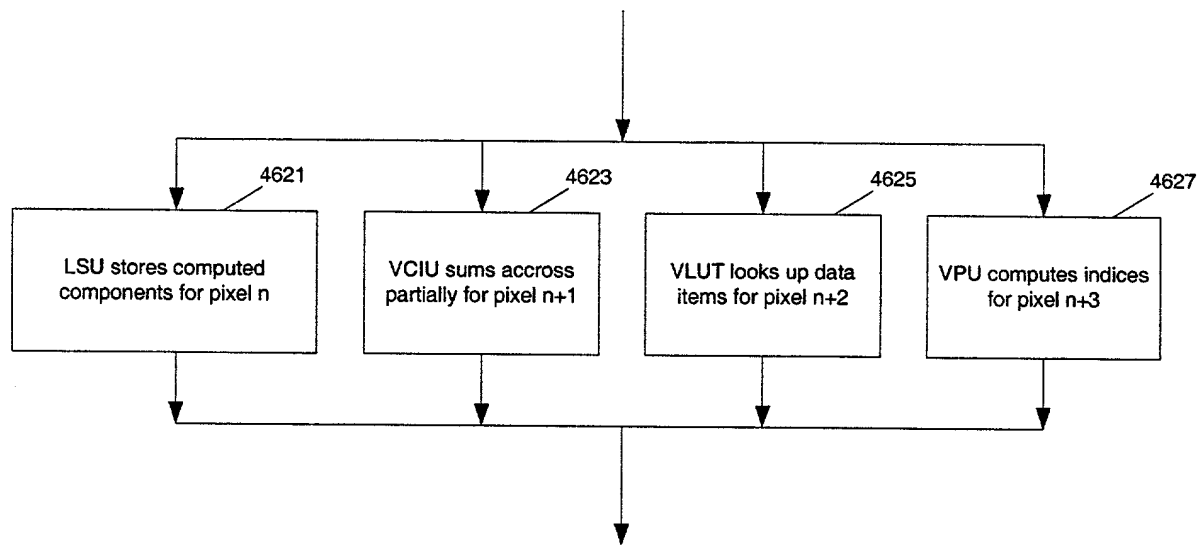


Fig. 93

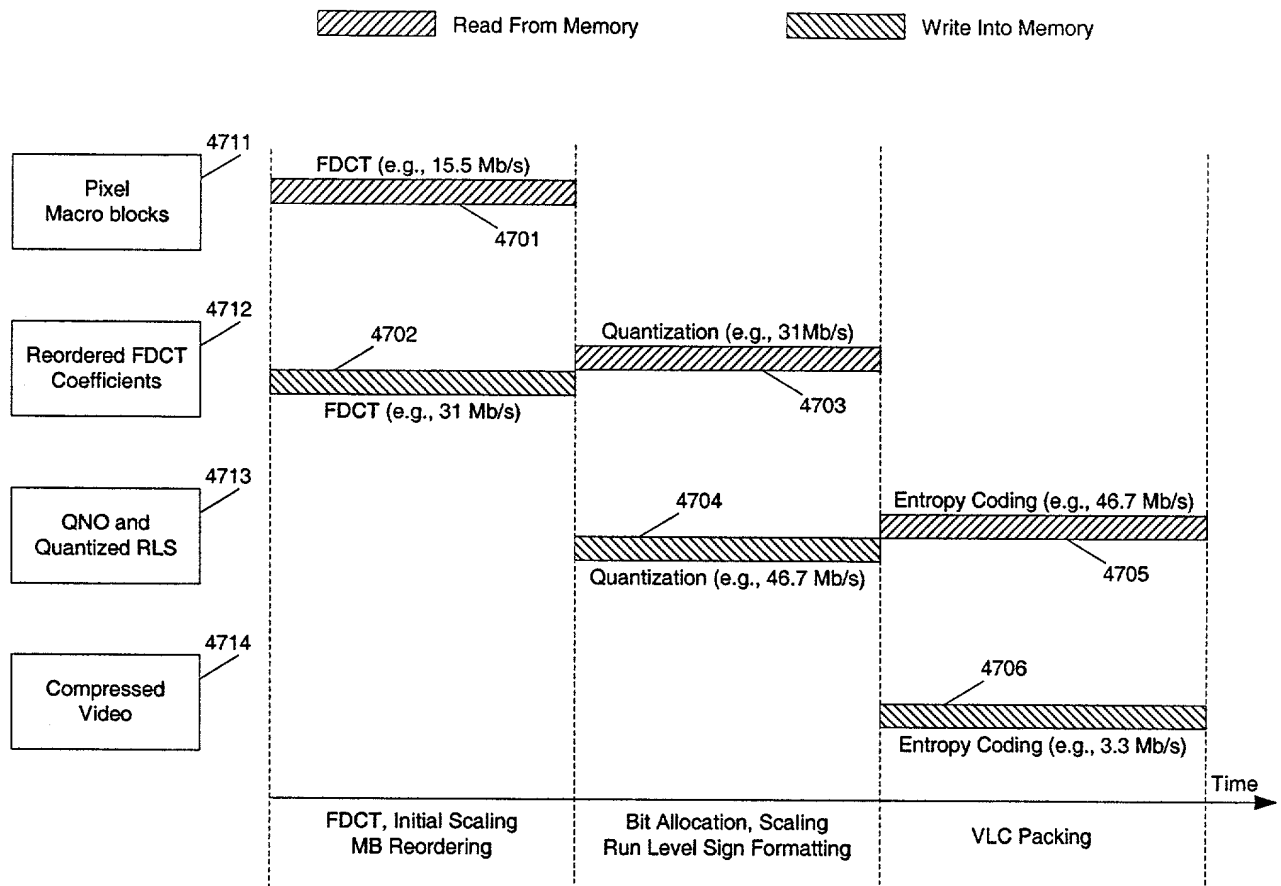


Fig. 94

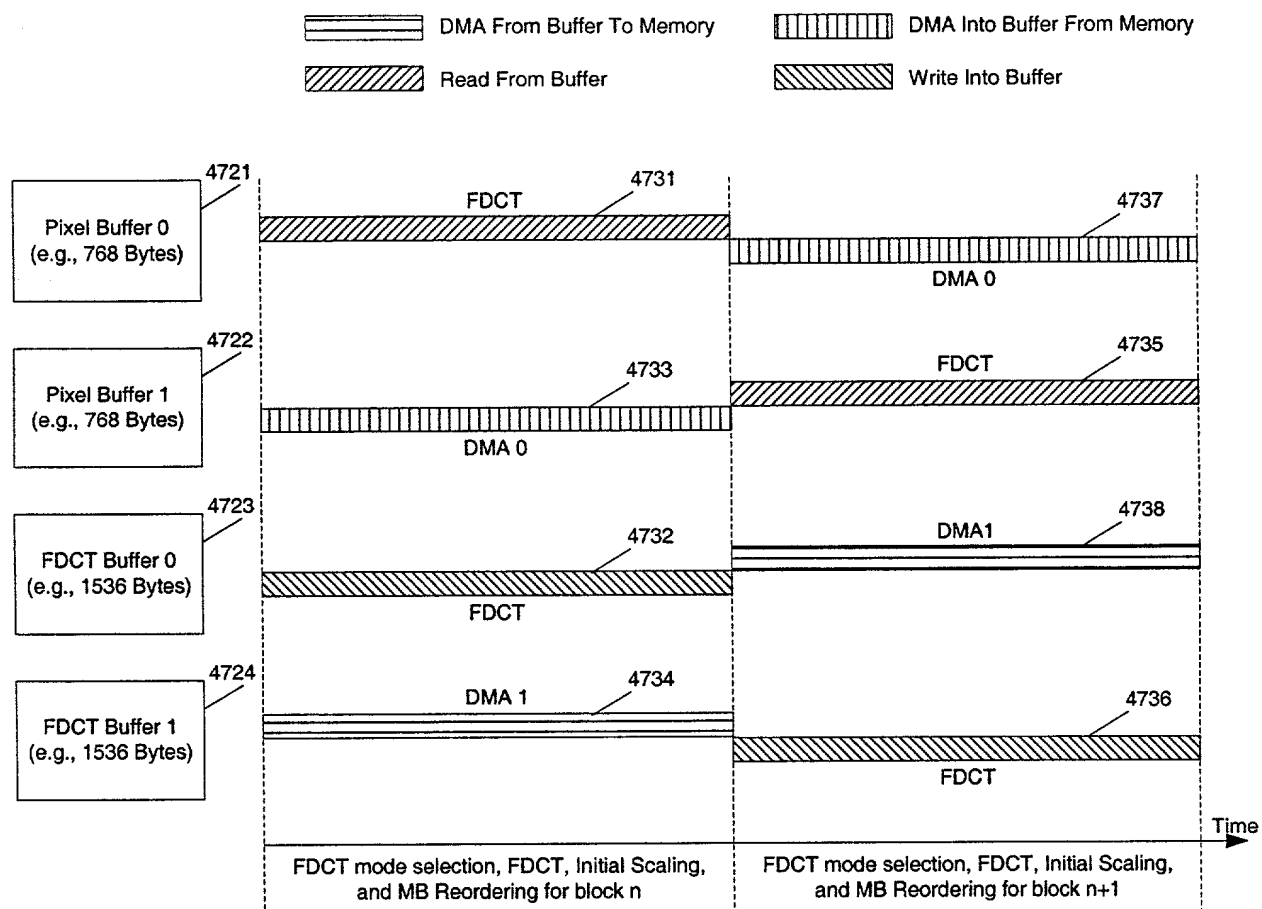


Fig. 95

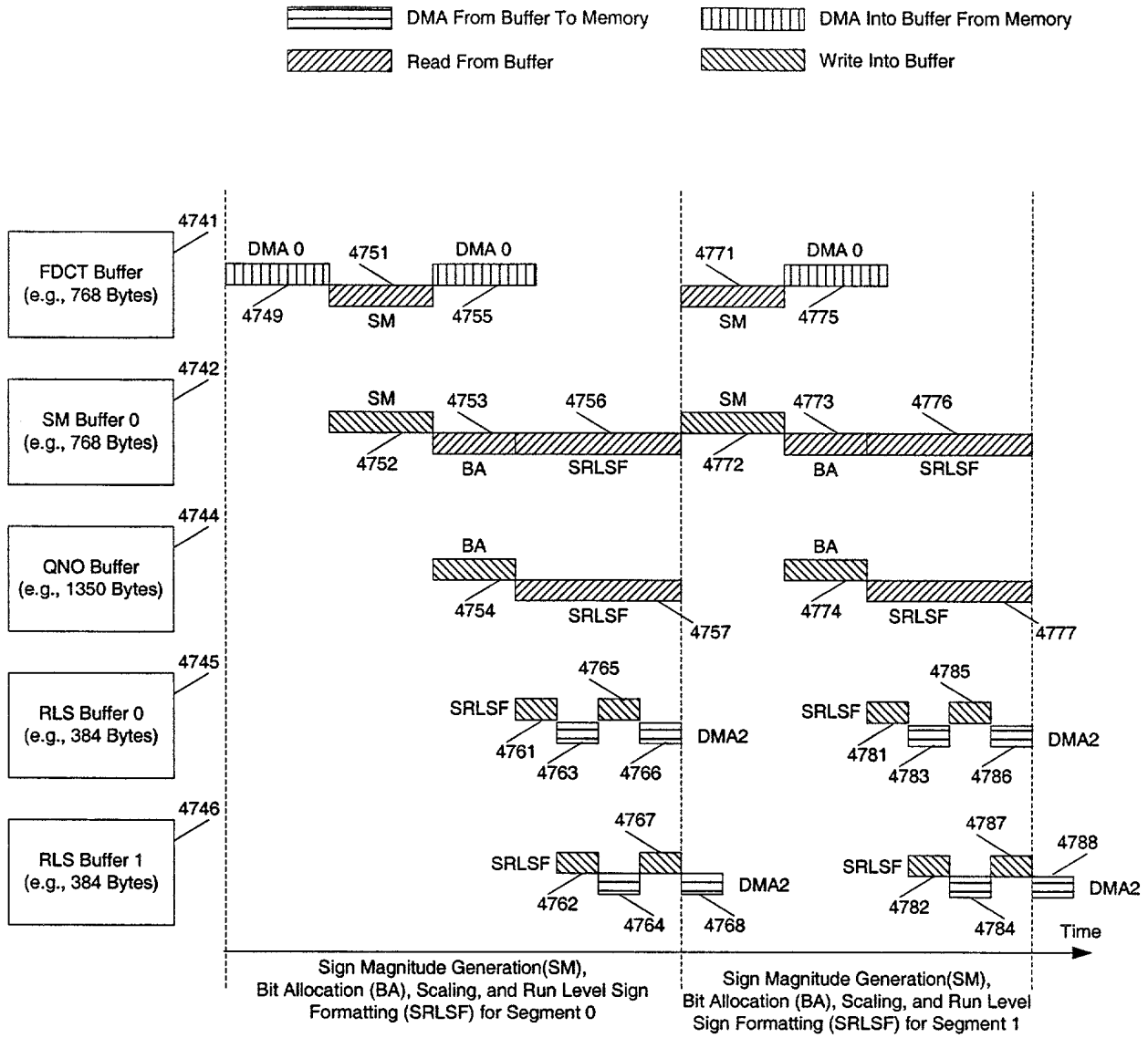


Fig. 96

The diagram illustrates the timing of data transfer for VLC Packing Segment 0. It features a vertical timeline on the left with six buffers: VLC Buffer 0 (4801), VLC Buffer 1 (4802), RLS Buffer 0 (4803), RLS Buffer 1 (4804), RLS Buffer 2 (4805), RLS Buffer 3 (4806), and RLS Buffer 4 (4807). The buffers are organized into three groups: VLC buffers (0 and 1), and RLS buffers (0 through 4). The RLS buffers are grouped into three sets of three buffers each, with the last set containing only two buffers (3 and 4). The diagram shows the sequence of data transfer for each buffer, with the following labels: pass 1 & 2, mb 1, mb 2, mb 3, mb 4, mb 5, pass 3, and DMA 1. The DMA 1 label is repeated for each buffer group. The diagram also shows the sequence of data transfer for each buffer, with the following labels: pass 1 & 2, mb 1, mb 2, mb 3, mb 4, mb 5, pass 3, and DMA 1. The DMA 1 label is repeated for each buffer group. The diagram also shows the sequence of data transfer for each buffer, with the following labels: pass 1 & 2, mb 1, mb 2, mb 3, mb 4, mb 5, pass 3, and DMA 1. The DMA 1 label is repeated for each buffer group.

Fig. 97